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 NX-OS Troubleshooting Guide* and the *Cisco MDS 9000 Family NX-OS System Messages Reference*.

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 aaa

To enable debugging for boot variables, use the **debug aaa** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug aaa {**all** | **conf-events** | **errors** | **events** | **mts**}

no debug aaa {**all** | **conf-events** | **errors** | **events** | **mts**}

Syntax Description

all	Enables all AAA debug options.
conf-events	Enables AAA configuration events debugging.
errors	Enables debugging for AAA errors.
events	Enables debugging for AAA events.
mts	Enables AAA transmit and receive MTS packets debugging.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modifications
1.3(1)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug aaa conf-events** command is issued:

```
switch# debug aaa conf-events
Nov 20 06:29:52 aaa: aaa_cleanup_session
Nov 20 06:29:52 aaa: mts_drop of request msg
Nov 20 06:29:52 aaa: Configured method local Succeeded
Nov 20 06:29:58 aaa: Src: 0x00000101/10886  Dst: 0x00000101/0  ID: 0x003
ize: 197 [REQ] Opc: 8402 (MTS_OPC_AAA_REQ) RR: 0x003A48F7  HA_SEQNO: 0x0
 TS: 0x9FC1C1234E7C  REJ:0  SYNC:0
Nov 20 06:29:58 aaa: 01 01 0C 00 00 00 00 00 00 00 00 00 00 00 02 01
Nov 20 06:29:58 aaa: 00 00 00 00 00 00 00 06 08 00 03 05 00 00 00 00
Nov 20 06:29:58 aaa: 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Nov 20 06:29:58 aaa: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Nov 20 06:29:58 aaa: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Nov 20 06:29:58 aaa: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Nov 20 06:29:58 aaa: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Nov 20 06:29:58 aaa: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Nov 20 06:29:58 aaa: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Nov 20 06:29:58 aaa: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Related Commands

Command	Description
aaa authentication login	Configures the authentication mode for a login.
no debug all	Disables all debugging.
show aaa authentication	Displays the configured authentication methods.

debug all

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

debug all

no debug all

Syntax Description

This command has no arguments or keywords.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

The **no debug all** command turns off all diagnostic output. Using the **no debug all** command is a convenient way to ensure that you have not accidentally left any debug commands turned on.

Caution Because debugging output takes priority over other network traffic, and because the **debug all** command generates more output than any other **debug** command, it can severely diminish the performance of the switch or even render it unusable. In virtually all cases, it is best to use more specific **debug** commands.

Examples

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

```
switch# debug all
```

Related Commands

Command	Description
show debug	Displays the debug commands configured on the switch.

debug biosd

To configure bios_daemon debugging, use the **debug biosd** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug biosd all

no debug biosd all

Syntax Description

all	Enables all bios_daemon debug options.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
2.1(1)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug biosd
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug bootvar

To enable debugging for boot variables, use the **debug bootvar** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

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

no debug bootvar {**all** | **errors** | **events** | **info** | **pss**}

Syntax Description

all	Enables all boot variable debug options.
errors	Enables debugging for boot variable errors.
events	Enables debugging for boot variable events.
info	Enables debugging for boot variable information.
pss	Enables debugging for boot variable PSS operations.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug bootvar all
```

Related Commands

Command	Description
debug all	Enables debugging for all features on the switch.
show boot	Displays the boot variables or modules.

debug callhome

To enable debugging for the Call Home function, use the **debug callhome** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug callhome {**all** | **events** | **mts**}

no 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

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

The **debug callhome** command, when used with the **all** parameter, displays the troubleshooting information for both Call Home event traces and a dump of the messaging and transaction service (MTS) messages that the Call Home function receives.

Note The debug Call Home function displays event traces for both successful and unsuccessful Call Home e-mail transmissions.

Examples

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

```
switch# debug callhome events
2005-03-09T05:37:21 2005 Mar  9 05:37:21 callhome: filling in name field with Test
2005 Mar  9 05:37:21 callhome: filling in the header list
2005 Mar  9 05:37:21 callhome: filling up the chassis list
2005 Mar  9 05:37:21 callhome: filling up the main body list
2005 Mar  9 05:37:21 callhome: filling up the fru list 2005 Mar  9 05:37:21 callhome:
Entering function do_event_correlation
2005 Mar  9 05:37:21 callhome: getting dest profiles for alert group test
2005 Mar  9 05:37:21 callhome: getting dest profiles for alert group cisco-tac
2005 Mar  9 05:37:21 callhome: Applying the event rule for destination profile full_txt
2005 Mar  9 05:37:21 callhome: Applying the event rule for destination profile short_txt
2005 Mar  9 05:37:21 callhome: Applying the event rule for destination profile xml 2005
Mar  9 05:37:21 callhome: Applying the event rule for destination profile basu
2005 Mar  9 05:37:21 callhome: Exiting function do_event_correlation
```

```
2005 Mar  9 05:37:21 callhome: running cli commands for alert name : Test, message id :
1540383426
2005 Mar  9 05:37:21 callhome: process scheduled for running cli commands for alert Test,
message id 1540383426, destination profile basu
2005 Mar  9 05:37:21 callhome: process scheduled for running cli commands for alert Test,
message id 1540383426, destination profile xml
2005 Mar  9 05:37:21 callhome: process scheduled for running cli commands for alert Test,
message id 1540383426, destination profile short_txt
.
```

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
...
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show callhome	Displays Call Home information configured on a switch.

debug cert-enroll

To enable debugging for the certificate enroll daemon, use the **debug cert-enroll** command in EXEC mode. To disable a **debug** command use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug cert-enroll {all | config | config-lowlevel | request | request-lowlevel}

no debug cert-enroll {all | config | config-lowlevel | request | request-lowlevel}

Syntax Description

all	Enables all debugging flags.
config	Enables debugging for the certificate enroll configuration.
config-lowlevel	Enables low-level debugging for the certificate enroll configuration.
request	Enables debugging for the certification enroll request.
request-lowlevel	Enables low-level debugging for the certification enroll request.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
3.0(1)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug cert-enroll all** command is issued:

```
switch# debug cert-enroll all
2006 Jan 21 00:44:52.875125 cert_enroll: cert_en_debug_conf_open: entering...
2006 Jan 21 00:44:52.875602 cert_enroll: cert_en_debug_conf_open: exiting
2006 Jan 21 00:44:52.876284 cert_enroll: cert_en_conf_close: entering...
2006 Jan 21 00:44:52.876349 cert_enroll: cert_en_conf_close: returning 0
2006 Jan 21 00:44:52.876400 cert_enroll: cert_en_enable_info_config: entering for
Cert-enroll Daemon debug
2006 Jan 21 00:44:52.876428 cert_enroll: cert_en_debug_conf_open: entering...
2006 Jan 21 00:44:52.876679 cert_enroll: cert_en_debug_conf_open: exiting
sw-46-180# 2006 Jan 21 00:44:52.876712 cert_enroll: cert_en_enable_info_config:
SET_REQ for Cert-enroll Daemon debug with 1
2006 Jan 21 00:44:52.876857 cert_enroll: cert_en_enable_info_config: SET_REQ done for
Cert-enroll Daemon debug with 1
2006 Jan 21 00:44:52.876896 cert_enroll: cert_en_enable_info_config: got back the return
value of configuration operation:success
2006 Jan 21 00:44:52.876922 cert_enroll: cert_en_debug_conf_close: entering...
2006 Jan 21 00:44:52.876965 cert_enroll: cert_en_debug_conf_close: returning 0
2006 Jan 21 00:44:52.876991 cert_enroll: cert_en_enable_info_config: exiting for
 Cert-enroll Daemon debug...
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show crypto ca certificates	Displays configured trust point certificates.

debug cdp

To enable debugging for the Cisco Discovery Protocol (CDP) function, use the **debug cdp** command in EXEC mode. To disable a **debug** command use the **no** form of the 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**}]

no debug cdp {**all** | **errors** | **events** {**mts** | **packets** | **pss**}} [**interface** {**gigabitethernet** *slot/port* | **mgmt 0**}]

Syntax Description

all	Enables debugging for all CDP features.
errors	Enables debugging for CDP error conditions.
events	Enables debugging for CDP events.
mts	Enables debugging for CDP tx/rx MTS packets.
packets	Enables debugging for CDP tx/rx CDP packets.
pss	Enables debugging for all PSS related CDP events.
interface	(Optional) Specifies debugging for the specified interface.
gigbitethernet *slot/port*	Specifies the Gigabit Ethernet interface slot and port.
mgmt 0	Specifies the management interface.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.1(1)	This command was introduced.

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
...
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show cdp	Displays CDP parameters configured globally or for a specific interface.

debug cfs

To enable debugging for Cisco Fabric Services (CFS), use the **debug cfs** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug cfs {**all** | **errors** | **events** {**db** [**vsan** *vsan-id*] | **fc2** [**vsan** *vsan-id*] | **fsm-action** [**vsan** *vsan-id*] | **fsm-trans** [**sap** *sap-id*] | **mts** [**vsan** *vsan-id*] | **pss** [**vsan** *vsan-id*]} | **fsm** {**ha** | **trans**} | **merge**}

no debug cfs {**all** | **errors** | **events** {**db** [**vsan** *vsan-id*] | **fc2** [**vsan** *vsan-id*] | **fsm-action** [**vsan** *vsan-id*] | **fsm-trans** [**sap** *sap-id*] | **mts** [**vsan** *vsan-id*] | **pss** [**vsan** *vsan-id*]} | **fsm** {**ha** | **trans**} | **merge**}

Syntax Description

all	Enables all CFS debugging.
errors	Enables debugging for CFS error conditions.
events	Enables debugging for CFS events.
db	Enables debugging for CFS database events.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN ID. The range is 1 to 4093.
fc2	Enables debugging for CFS FC2 events.
fsm-action	Enables debugging for CFS FSM action events.
fsm-trans	Enables debugging for CFS FSM transition events.
sap *sap-id*	(Optional) Restricts debugging to the specified SAP ID. The range is 0 to 2147483647.
mts	Enables debugging for CFS MTS events.
pss	Enables debugging for CFS PSS events.
fsm	Enables debugging for CFS FSM events.
ha	Enables debugging for CFS FSM high availability events.
trans	Enables debugging for CFS FSM transition events.
merge	Enables debugging for CFS merge events.

Defaults

None.

Command Modes

EXEC mode.

Command History

Release	Modification
2.0(x)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug cfs all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show cfs	Displays CFS information.

debug cimserver

To enable debugging for the Common Information Model (CIM) management applications function, use the **debug cimserver** command in EXEC mode. To disable a debug command use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug cimserver {**all** | **errors** | **events** | **mts** | **trace**}

no debug cimserver {**all** | **errors** | **events** | **mts** | **trace**}

Syntax Description

all	Enables debugging for all CIM features.
errors	Enables debugging for CIM error conditions.
events	Enables debugging for CIM events.
mts	Enables debugging for CIM tx/rx MTS packets.
trace	Enables debugging for CIM traces.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.3(1)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug cimserver all
2004 Mar 29 20:05:22 cimsrvprov: cim_mts_dispatch(): Opcode is 182
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show cimserver	Displays the CIM configurations and settings.

debug cloud

To enable debugging of cloud discovery, use the **debug cloud** command in EXEC mode.To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug cloud {**all | bypass ficon_mgr | cloud | conditional | demux vsan** *vsan-id* **| deque | discovery | error | event vsan** *vsan-id* **| ha vsan** *vsan-id* **| init | member | memory | messages | remotesync | trace** [**detail vsan** *vsan-id* **| vsan** *vsan-id*] **| warning** [*vsan-id*] **| xipc**}

no debug cloud {**all | bypass ficon_mgr | cloud | conditional | demux vsan** *vsan-id* **| deque | discovery | error | event vsan** *vsan-id* **| ha vsan** *vsan-id* **| init | member | memory | messages | remotesync | trace [detail vsan** *vsan-id* **| vsan** *vsan-id*] **| warning [***vsan-id*] **| xipc**}

Syntax Description

all	Enables debugging of all features of the cloud.
bypass	Enables some components in cloud execution to be bypassed during debugging.
ficon_mgr	Enables the FICON manager to be bypassed during debugging.
cloud	Enables debugging of all cloud commands.
conditional	Enables debugging of the cloud discovery conditional service.
demux	Enables debugging of the cloud message demux.
vsan *vsan-id*	Restricts debugging to the specified VSAN ID. The range is 1 to 4094.
deque	Enables debugging of the cloud message dequeue.
discovery	Enables debugging of the discovery process.
error	Enables debugging of the cloud errors.
event vsan	Enables debugging of the cloud finite state machine (FSM) and events.
ha vsan	Enables debugging of cloud high availability (HA).
init	Enables debugging of cloud discovery initialization.
member	Enables debugging of cloud member changes.
memory	Enables debugging of cloud memory allocation.
messages	Enables debugging of cloud discovery messaging and transaction service (MTS) messages.
remotesync	Enables debugging of discovery remote sync.
trace	Enables debugging of the cloud trace.
detail	(Optional) Enables debugging of the cloud detailed trace.
warning	Enables debugging of cloud warnings.
xipc	Enables debugging of XIPC messages.

Defaults

None.

Command Modes

EXEC mode.

Command History

Release	Modification
3.0(1)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays system output from the **debug cloud all** command:

```
switch# debug cloud all
1980 Feb 15 22:03:41.650721 cloud: fu_fsm_execute_all: match_msg_id(0), log_alre
ady_open(0)
1980 Feb 15 22:03:41.650874 cloud: fu_fsm_execute_all: null fsm_event_list
1980 Feb 15 22:03:41.650956 cloud: fu_fsm_engine_post_event_processing: mts msg
MTS_OPC_DEBUG_WRAP_MSG(msg_id 1302150) dropped
1980 Feb 15 22:03:41.651000 cloud: cloud_deque
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show cloud discovery	Displays cloud discovery information.
show cloud membership	Displays information about members of the cloud.

debug core

To enable core daemon debugging, use the **debug core** command in EXEC mode. To disable a debug command use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug core {**error** | **flow**}

no debug core {**error** | **flow**}

Syntax Description

error	Enables debugging for core demon error conditions.
flow	Enables debugging for the core demon flow.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug core flow
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show cores	Displays all the cores presently available for upload from active supervisor.

debug device-alias

To enable debugging for device aliases, use the **debug device-alias** command in EXEC mode. To disable a **debug** command use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug device-alias {**all** | **database** {**detail** | **errors** | **events**} | **fsm** | **ha** | **import** {**errors** | **events**} | **merge** {**errors** | **events** | **packets**} | **pss** {**errors** | **events**} | **session** {**errors** | **events** | **packets**} | **trace**}

no debug device-alias {**all** | **database** {**detail** | **errors** | **events**} | **fsm** | **ha** | **import** {**errors** | **events**} | **merge** {**errors** | **events** | **packets**} | **pss** {**errors** | **events**} | **session** {**errors** | **events** | **packets**} | **trace**}

Syntax Description

all	Enables all device alias debugging.
database	Enables debugging for device alias database events.
detail	Enables detailed debugging for device alias database events.
errors	Enables debugging for device alias error conditions.
events	Enables debugging for device alias events.
fsm	Enables debugging for device alias FSM events.
ha	Enables debugging for device alias HA events.
import	Enables debugging for device alias imports.
merge	Enables debugging for device alias merges.
packets	Enables debugging for device alias packets.
pss	Enables debugging for device alias PSS.
session	Enables debugging for device alias sessions.
trace	Enables debugging for device alias traces.

Defaults

None.

Command Modes

EXEC mode.

Command History

Release	Modification
2.0(x)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug device-alias all** command is issued:

```
switch# debug device-alias all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show device-alias	Displays device alias information.

debug dpvm

To enable debugging for dynamic port VSAN membership (DPVM), use the **debug dpvm** command in EXEC mode. To disable a **debug** command use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug dpvm {**all** | **cfs-events** | **change-events** | **db-events** | **errors** | **ftrace** | **merge-event** | **mts-events** | **pss-events** | **session-events** | **snmp-events** | **sys-events**}

no debug dpvm {**all** | **cfs-events** | **change-events** | **db-events** | **errors** | **ftrace** | **merge-event** | **mts-events** | **pss-events** | **session-events** | **snmp-events** | **sys-events**}

Syntax Description

all	Enables debugging for all DPVM.
cfs-events	Enables debugging for Cisco Fabric Services (CFS).
change-events	Enables debugging for change events.
db-events	Enables debugging for database events.
errors	Enables debugging for error.
ftrace	Enables debugging for function trace.
merge-event	Enables debugging for merge events.
mts-events	Enables debugging for MTS events.
pss-events	Enables debugging for PSS events.
session-events	Enables debugging for session events.
snmp-events	Enables debugging for SNMP events.
sys-events	Enables debugging for system events.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
2.0(x)	This command was introduced.

Usage Guidelines

To use this command, DPVM must be enabled using the **dpvm enable** command.

Examples

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

```
switch# debug dpvm all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show dpvm	Displays DPVM database information.

debug dstats

To enable delta statistics debugging, use the **debug dstats** command in EXEC mode. To disable a **debug** command use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug dstats {**error** | **flow**}

no debug dstats {**error** | **flow**}

Syntax Description

error	Enables debugging for delta statistics error conditions.
flow	Enables debugging for the delta statistics flow.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug dstats flow
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug ethport

To enable Ethernet port debugging, use the **debug ethport** command in EXEC mode. To disable a **debug** command, use the **no** form of the 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 gigibetethernet** *slot/port* | **module** *slot*] | **trace** [**interface gigibetethernet** *slot/port* | **module** *slot*]}

no debug ethport {**all | error | event** [**interface gigabitethernet** *slot/port* | **module** *slot*] | **ha** [**interface gigibetethernet** *slot/port* | **module** *slot*] | **trace** [**interface gigibetethernet** *slot/port* | **module** *slot*]}

Syntax Description

all	Enables debugging for all Ethernet port features.
error	Enables debugging for Ethernet port error conditions.
event	Enables debugging for Ethernet port events.
ha	Enables debugging for port high availability.
trace	Enables debugging for Ethernet port traces.
interface gigibetethernet *slot/port*	(Optional) Specifies the slot and port of the Gigabit Ethernet interface.
module *slot*	(Optional) Specifies the slot number of the module being debugged.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug ethport all
1981 May  5 07:28:59 ethport: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
1981 May  5 07:28:59 ethport: fu_fsm_execute_all: null fsm_event_list
1981 May  5 07:28:59 ethport: fu_fsm_engine_post_event_processing: mts msg
MTS_OPC_DEBUG_WRAP_MSG(msg_id 52343) dropped
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug exceptionlog

To enable the exception log debugging feature, use the **debug exceptionlog** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

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

no debug exceptionlog {**demux** | **deque** | **error** | **flow** | **info**}

Syntax Description

demux	Enables debugging for the exception logger demux functions.
deque	Enables debugging for the exception logger deque function.
error	Enables debugging for exception logger errors.
flow	Enables debugging for the exception logger flow.
info	Enables debugging for exception logger information.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug fabric-binding

To enable debugging for the fabric binding feature, use the **debug fabric-binding** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fabric-binding {**all** | **efmd** {**db-events** | **errors** | **merge** {**errors** | **events** | **packets**}} | **mts-events** | **pss-events**} | **errors** [**vsan** *vsan-id*] | **events** [**vsan** *vsan-id*] | **mts-events** | **pss-events** | **snmp-events** | **trace** [**vsan** *vsan-id*]}

no debug fabric-binding {**all** | **efmd** {**db-events** | **errors** | **merge** {**errors** | **events** | **packets**}} | **mts-events** | **pss-events**} | **errors** [**vsan** *vsan-id*] | **events** [**vsan** *vsan-id*] | **mts-events** | **pss-events** | **snmp-events** | **trace** [**vsan** *vsan-id*]}

Syntax Description

all	Enables debugging for all fabric binding features.
db-events	Enables debugging for EFMD protocol database events.
efmd	Enables debugging for Exchange Fabric Membership Data (EFMD) protocol.
errors	Enables debugging for fabric binding errors.
events	Enables debugging for fabric binding events.
merge	Enables debugging for EFMD protocol merges.
packets	Enables debugging for EFMD protocol packets.
vsan *vsan-id*	(Optional) Specifies the VSAN ID. The range is 1 to 4093.
events	Enables debugging for fabric binding events.
mts-events	Enables debugging for fabric binding MTS events.
pss-events	Enables debugging for fabric binding PSS events.
snmp-events	Enables debugging for fabric binding SNMP events
trace	Enables debugging for fabric binding traces.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.3(2)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug fabric-binding all** command is issued:

```
switch# debug fabric-binding all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show fabric-binding	Displays configured fabric binding information.

debug fc-tunnel

To enable debugging for the Fibre Channel tunnel feature, use the **debug fc-tunnel** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fc-tunnel {**all** | **errors** | **external-events** | **ha** | **label-update** | **mts** {**pkt** | **pkthdr**} {**both** | **rx** | **tx**} | **pss** | **route-update** [**vsan** *vsan-id*] | **rsvp-messages** [**tunnel** *tunnel-id* | **vsan** *vsan-id*] | **state-machine**}

no debug fc-tunnel {**all** | **errors** | **external-events** | **ha** | **label-update** | **mts** {**pkt** | **pkthdr**} {**both** | **rx** | **tx**} | **pss** | **route-update** [**vsan** *vsan-id*] | **rsvp-messages** [**tunnel** *tunnel-id* | **vsan** *vsan-id*] | **state-machine**}

Syntax Description

all	Enables debugging for all FC tunnel features.
errors	Enables debugging for FC tunnel errors.
external-events	Enables debugging for external FC tunnel events.
ha	Enables debugging for FC tunnel high availability (HA) events.
label-update	Enables debugging for FC tunnel label updates.
mts	Enables debugging for FC tunnel MTS events.
pkt	Specifies debugging of packets.
pkthdr	Specifies debugging of headers.
both	Specifies debugging in both the transmit and receive directions.
tx	Specifies debugging in the transmit direction.
rx	Specifies debugging in the receive direction.
pss	Enables debugging for FC tunnel PSS events.
route-update	Enables debugging for FC tunnel route updates.
vsan *vsan-id*	(Optional) Specifies the VSAN ID. The range is 1 to 4093.
rsvp-messages	Enables debugging for FC tunnel SNMP events
tunnel *tunnel-id*	(Optional) Specifies the tunnel ID. The range is 1 to 255.
state-machine	Enables debugging for FC tunnel traces.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.3(2)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug fc-tunnel all** command is issued:

```
switch# debug fc-tunnel all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show fc-tunnel	Display configured FC tunnel information.

debug fc2

To enable debugging for the FC2 feature, use the **debug fc2** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fc2 {**credit** | **error** [**fcid** *fcid* [**interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] | **interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] **flag** | **flow** [**fcid** *fcid* [**interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] | **interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] | **(interface fc** *type number* | **vsan** *vsan-id***)** | **frame** | **loopback pkt** {**both** | **tx** | **rx**} [**bytes** *bytes* | **fcid** *fcid* [**bytes** *bytes* | **interface** {**fc** *slot/port* | **fcip** *port*} [**bytes** *bytes* | **pkts** *pkts* [**bytes** *bytes*]] | **pkts** *pkts* [**bytes** *bytes*] | **vsan** *vsan-id* [**bytes** *bytes* | **interface** {**fc** *slot/port* | **fcip** *port*} [**bytes** *bytes* | **pkts** *pkts* [**bytes** *bytes*]]] | **pkthdr** {**both** | **tx** | **rx**} [**bytes** *bytes* | **fcid** *fcid* [**bytes** *bytes* | **interface** {**fc** *slot/port* | **fcip** *port*} [**bytes** *bytes* | **pkts** *pkts* [**bytes** *bytes*]] | **pkts** *pkts* [**bytes** *bytes*] | **vsan** *vsan-id* [**bytes** *bytes* | **interface** {**fc** *slot/port* | **fcip** *port*} [**bytes** *bytes* | **pkts** *pkts* [**bytes** *bytes*]]] | **rdl** | **rxhdrhistory** [**fcid** *fcid* [**interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] | **interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] **txhdrhistory** [**fcid** *fcid* [**interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] | **interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]]}

no debug fc2 {**credit** | **error** [**fcid** *fcid* [**interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] | **interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] **flag** | **flow** [**fcid** *fcid* [**interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] | **interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] | **(interface fc** *type number* | **vsan** *vsan-id***)** | **frame** | **loopback** | **pkt** {**both** | **tx** | **rx**} [**bytes** *bytes* | **fcid** *fcid* [**bytes** *bytes* | **interface** {**fc** *slot/port* | **fcip** *port*} [**bytes** *bytes* | **pkts** *pkts* [**bytes** *bytes*]] | **pkts** *pkts* [**bytes** *bytes*] | **vsan** *vsan-id* [**bytes** *bytes* | **interface** {**fc** *slot/port* | **fcip** *port*} [**bytes** *bytes* | **pkts** *pkts* [**bytes** *bytes*]]] | **pkthdr** {**both** | **tx** | **rx**} [**bytes** *bytes* | **fcid** *fcid* [**bytes** *bytes* | **interface** {**fc** *slot/port* | **fcip** *port*} [**bytes** *bytes* | **pkts** *pkts* [**bytes** *bytes*]] | **pkts** *pkts* [**bytes** *bytes*] | **vsan** *vsan-id* [**bytes** *bytes* | **interface** {**fc** *slot/port* | **fcip** *port*} [**bytes** *bytes* | **pkts** *pkts* [**bytes** *bytes*]]] | **rdl** |**rxhdrhistory** [**fcid** *fcid* [**interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] | **interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] **txhdrhistory** [**fcid** *fcid* [**interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]] | **interface** {**fc** *slot/port* | **fcip** *port*} | **vsan** *vsan-id* [**interface** {**fc** *slot/port* | **fcip** *port*}]]}

Syntax Description

credit	Enables FC2 credit debugging.
error	Enables FC2 error debugging.
fcid *fcid*	(Optional) Restricts debugging to the specified FCID.
interface	(Optional) Restricts debugging to the specified interface.
fc *slot/port*	(Optional) Restricts debugging to the specified interface.
fcip *port*	(Optional) Restricts debugging to the specified interface.
vsan *vsan-id*	Restricts debugging to the specified VSAN.
flag	Enables FC2 flags debugging.
flow	Enables FC2 flow debugging.
frame	Enables FC2 frame debugging.

loopback	Enables FC2 loopback debugging.
pkt	Enables FC packet debugging.
both	Enables debugging in both the transmit and receive directions.
tx	Enables debugging in the transmit direction,
rx	Enables debugging in the receive direction.
bytes *bytes*	(Optional) Specifies the number of bytes to display.
pkts *pkts*	Specifies the number of packets to display.
pkthdr	Enables FC header debugging.
rdl	Enables FC2 RDL debugging.
rxhdrhistory	Enables FC2 received header history debugging.
txhdrhistory	Enables FC2 transmitted header history debugging.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show fc2	Displays FC2 information.

debug fc2d

To enable debugging for the FC2 feature, use the **debug fc2** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fc2 {**all** | **bypass ficon_mgr** | **demux** [**vsan** *vsan-id*] | **deque** | **error** | **event** [**vsan** *vsan-id*] | **ha** [**vsan** *vsan-id*] | **trace** [**detail**] [**vsan** *vsan-id*] | **warning** [**vsan** *vsan-id*]}

no debug fc2 {**all** | **bypass ficon_mgr** | **demux** [**vsan** *vsan-id*] | **deque** | **error** | **event** [**vsan** *vsan-id*] | **ha** [**vsan** *vsan-id*] | **trace** [**detail**] [**vsan** *vsan-id*] | **warning** [**vsan** *vsan-id*]}

Syntax Description

all	Enables all FC2D debug flags.
bypass	Enables bypassing some components in fc2d execution.
ficon_mgr	Enables bypassing FICON Manager in fc2d execution.
demux	Enables debugging of FC2D message demux.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.
deque	Enables debugging of FC2D message dequeue.
error	Enables debugging of FC2D error.
event	Enables debugging of FC2D FSM and events.
ha	Enables debugging of FC2D HA.
trace	Enables debugging of FC2D trace.
detail	(Optional) Enables detailed debugging of FC2D trace.
warning	Enables debugging of FC2D warning.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.3(4)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug fc2d all** command is issued:

```
switch1# debug fc2d all
2004 Mar 29 22:57:25 fc2d: fu_fsm_execute_all: match_msg_id(0), log_already_open (0)
2004 Mar 29 22:57:25 fc2d: fu_fsm_execute_all: null fsm_event_list
2004 Mar 29 22:57:25 fc2d: fu_fsm_engine_post_event_processing: mts msg MTS_OPC_
DEBUG_WRAP_MSG(msg_id 6894921) dropped
```

Related Commands

Command	Description
debug all	Enables debugging for the FC2 feature.
no debug all	Disables all debugging.
show fc2	Displays FC2 information.

debug fcc

To enable debugging for the Fibre Channel Congestion (FCC) function, use the **debug fcc** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fcc {**all** | **error** [**module** *slot*] | **event** [**module** *slot*] | **mts** [**pkt** {**both** | **rx** [**node** *range* | **opcode** *range* | **sap** *range*] | **tx**} | **pkthdr** {**both** | **tx** | **rx** [**numpkt** *range*]} | **trace** [**module** *slot*]}

no debug fcc {**all** | **error** [**module** *slot*] | **event** [**module** *slot*] | **mts** {**pkt** {**both** | **rx** [**node** *range* | **opcode** *range* | **sap** *range*] | **tx**} | **pkthdr** {**both** | **tx** | **rx** [**numpkt** *range*]} | **trace** [**module** *slot*]}

Syntax Description

all	Enables debugging for all FCC features.
errors	Enables debugging for FCC error conditions.
events	Enables debugging for FCC events.
module *slot*	(Optional) Specifies the slot number of the module being debugged.
mts	Enables debugging for FCC tx/rx MTS packets.
pkt	Enables debugging for FCC tx/rx FCC packets.
both	Specifies debugging in both the transmit and receive directions.
tx	Specifies debugging in the transmit direction,
rx	Specifies debugging in the receive direction.
node *range*	(Optional) Specifies the node for the packets in the receive direction.
opcode *range*	(Optional) Specifies the opcode for the packets in the receive direction.
sap *range*	(Optional) Specifies the sap for the packets in the receive direction. The integer range is from 1 to 4096.
pkthdr	Enables debugging for FCC tx/rx FCC headers.
numpkt *range*	(Optional) Specifies the number of required packets
trace	Enables debugging for FCC traces.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug fcc all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show fcc	Displays FCC settings.

debug fcdomain

To enable debugging for the fcdomain feature, use the **debug fcdomain** command in EXEC mode.To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fcdomain {**all** | **critical** | **error** | **fc** {**pkt** | **pkthdr**} {**both** | **rx** | **tx**} [**interface** *type number* [**vsan** *vsan-id*] | **vsan** *vsan-id*] | **ipc** {**pkt** | **pkthdr**} {**both** | **rx** [**node** *range* | **opcode** *range* | **sap** *range*] | **tx**} | **memory** | **notify** | **phase**}

no debug fcdomain {**all** | **critical** | **error** | **fc** {**pkt** | **pkthdr**} {**both** | **rx** | **tx**} [**interface** *type number* [**vsan** *vsan-id*] | **vsan** *vsan-id*] | **ipc** {**pkt** | **pkthdr**} {**both** | **rx** [**node** *range* | **opcode** *range* | **sap** *range*] | **tx**} | **memory** | **notify** | **phase**}

Syntax Description

all	Enables debugging of all fcdomain parameters.
critical	Enables debugging of critical operations.
error	Enables debugging of error operation.
fc	Enables debugging of Fibre Channel packets and headers.
ipc	Enables debugging of Fibre Channel IP packets and headers.
pkt	Enables debugging of packets.
pkthdr	Enables debugging of headers.
both	Enables debugging in both the transmit and receive directions.
rx	Enables debugging in the receive direction.
tx	Enables debugging in the transmit direction.
interface *type number*	(Optional) Specifies the interface to be debugged.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.
node *range*	(Optional) Specifies the node for the packets in the receive direction.
opcode *range*	(Optional) Specifies the opcode for the packets in the receive direction.
sap *range*	(Optional) Specifies the sap for the packets in the receive direction. The integer range is from 1 to 4096.
memory	Enables debugging of memory operations.
notify	Enables debugging of notifications.
phase	Enables debugging of global phases.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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:

```
switch# 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
fcdomain	Enables fcdomain features.
show fcdomain domain-list	Displays current domains in the fabric.

debug fcfwd

To enable debugging for the Fibre Channel forwarding feature, use the **debug fcfwd** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fcfwd {**flogimap** | **idxmap** | **pcmap** | **sfib** | **spanmap**} {**error** | **event** | **trace**} [**module** *slot* | **vsan** *vsan-id* [**module** *slot*]]

no debug fcfwd {**flogimap** | **idxmap** | **pcmap** | **sfib** | **spanmap**} {**error** | **event** | **trace**} [**module** *slot* | **vsan** *vsan-id* [**module** *slot*]]

Syntax Description

flogimap	Enables flogimap debugging.
idxmap	Enables idxmap debugging.
pcmap	Enables pcmap debugging.
sfib	Enables sfib debugging.
spanmap	Enables spanmap debugging.
error	Enables debugging for FCC error conditions.
event	Enables debugging for FCC events.
trace	Enables debugging for FCC traces.
module *slot*	(Optional) Specifies the slot number of the module being debugged.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug fcfwd error
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show fcfwd	Displays the configured fcfwd tables and statistics.

debug fcns

To enable debugging for name server registration, use the **debug fcns** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fcns {**all** | **errors** | **events** {**mts** | **query** | **register**}} [**vsan** *vsan-id*]

no debug fcns {**all** | **errors** | **events** {**mts** | **query** | **register**}} [**vsan** *vsan-id*]

Syntax Description

all	Enables debugging for all name server features.
errors	Enables debugging for name server error conditions.
events	Enables debugging for name server events.
mts	Enables debugging for name server tx/rx MTS packets.
query	Enables debugging for name server tx/rx CDP packets.
register	Enables debugging for name server PSS related events.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
...
```

Related Commands

Command	Description
no debug all	Disables all debugging.

Command	Description
show fcns database	Displays the results of the discovery or the name server database for a specified VSAN or for all VSANs.
show fcns statistics	Displays the statistical information for a specified VSAN or for all VSANs.

debug fcs

To enable debugging for the fabric configuration server, use the **debug fcs** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fcs {**all** | **discovery events** | **errors** [**vsan** *vsan-id*] | **ess-events** [**vsan** *vsan-id*] | **mts events** {**brief** | **detail**} | **pss events** | **queries events** [**vsan** *vsan-id*] | **registrations events** [**vsan** *vsan-id*] | **rscn events** [**vsan** *vsan-id*] | **snmp events**}

no debug fcs {**all** | **discovery events** | **errors** [**vsan** *vsan-id*] | **ess-events** [**vsan** *vsan-id*] | **mts events** {**brief** | **detail**} | **pss events** | **queries events** [**vsan** *vsan-id*] | **registrations events** [**vsan** *vsan-id*] | **rscn events** [**vsan** *vsan-id*] | **snmp events**}

Syntax Description

all	Enables debugging for all FCS features.
discovery events	Enables debugging for FCS discovery events.
errors	Enables debugging for FCS error conditions.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.
ess-events	Enables debugging for FCS tx/rx ESS events.
mts events	Enables debugging for FCS tx/rx MTS events.
brief	Provides brief information for each event.
detail	Provides detailed information for each event.
pss events	Enables debugging for FCS
queries events	Enables debugging for FCS tx/rx events.
registration events	Enables debugging for FCS PSS related events.
rscn events	Enables debugging for FCS RSCN events.
snmp events	Enables debugging for FCS SNMP events.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug fcs all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show fcs	Displays the status of the fabric configuration.

debug fcsp-mgr

To enable debugging for the Fibre Channel Security Protocol (FC-SP) manager, use the **debug fcsp-mgr** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fcsp-mgr {**all** | **critical** | **datastructure** | **dhchap** | **error** | **event-gen** | **fc2** | **fsm** | **general** | **ha** | **init** | **level1** | **level2** | **level3** | **level4** | **level5** | **message** | **mts** | **notify** | **trace**}

no debug fcsp-mgr {**all** | **critical** | **datastructure** | **dhchap** | **error** | **event-gen** | **fc2** | **fsm** | **general** | **ha** | **init** | **level1** | **level2** | **level3** | **level4** | **level5** | **message** | **mts** | **notify** | **trace**}

Syntax Description

all	Enables debugging for all FC-SP features.
critical	Enables debugging of FC-SP critical errors.
datastructure	Enables debugging of FC-SP data structures.
dhchap	Enables debugging of DHCHAP.
error	Enables debugging of FC-SP error.
event-gen	Enables debugging of FC-SP event generation.
fc2	Enables debugging of FC-SP FC2 messages.
fsm	Enables debugging of FC-SP events.
general	Enables general debugging of FC-SP.
ha	Enables debugging of FC-SP high availability
init	Enables debugging of FC-SP initialization.
level1	Sets debugging level of FC-SP Mgr to 1.
level2	Sets debugging level of FC-SP Mgr to 2.
level3	Sets debugging level of FC-SP Mgr to 3.
level4	Sets debugging level of FC-SP Mgr to 4.
level5	Set debugging level of FC-SP Mgr to 5.
message	Enables debugging of FC-SP messages.
mts	Enables debugging of FC-SP MTS messages.
notify	Sets debug level to notify.
trace	Enables debugging of FC-SP function enter/exit.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.3(2)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug fcsp-mgr all** command is issued:

```
switch# debug fcsp-mgr all
2004 Mar 29 23:33:56 fcsp-mgr: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
2004 Mar 29 23:33:56 fcsp-mgr: fu_fsm_execute_all: null fsm_event_list
2004 Mar 29 23:33:56 fcsp-mgr: fu_fsm_engine_post_event_processing: mts msg MTS_
OPC_DEBUG_WRAP_MSG(msg_id 7061762) dropped
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show fcsp	Displays the status of the FC-SP configuration

debug fdmi

To enable debugging for the Fabric-Device Management Interface (FDMI) feature, use the **debug fdmi** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fdmi {**all** | **errors** | **fdmi-messages** [**vsan** *vsan-id*] | **ha** | **mts** {**pkt** {**both** | **rx** [**node** *range* | **opcode** *range* | **sap** *range*] | **tx**} | **pkthdr** {**both** | **tx** | **rx** [**numpkt** *range*]} | **pss** | **trace**}

no debug fdmi {**all** | **errors** | **fdmi-messages** [**vsan** *vsan-id*] | **ha** | **mts** {**pkt** {**both** | **rx** [**node** *range* | **opcode** *range* | **sap** *range*] | **tx**} | **pkthdr** {**both** | **tx** | **rx** [**numpkt** *range*]} | **pss** | **trace**}

Syntax Description

all	Enables debugging for all FDMI features.
errors	Enables debugging for FDMI error conditions.
fdmi-messages	Enables the dump of FDMI PDUs.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.
ha	Enables the dump of HA synchronization messages.
mts	Enables debugging for FDMI tx/rx MTS events.
pkt	Enables debugging for FCC tx/rx FCC packets.
both	Specifies debugging in both the transmit and receive directions.
tx	Specifies debugging in the transmit direction,
node *range*	(Optional) Specifies the node for the packets in the receive direction. The integer range is from 1 to 4096.
opcode *range*	(Optional) Specifies the opcode for the packets in the receive direction. The integer range is from 1 to 4096.
sap *range*	(Optional) Specifies the sap for the packets in the receive direction. the integer range is from 1 to 4096.
rx	Specifies debugging in the receive direction.
pkthdr	Enables debugging for FCC tx/rx FCC headers.
numpkt *range*	Specifies the number of required packets
pss	Enables debugging for FDMI PSSs.
trace	Restricts debugging for FDMI traces.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.3(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug fdmi all
2005 Mar 10 02:37:28 fdmi: 00 00 00 02 00 00 00 1C 04 19 65 08 00 82 39 08
2005 Mar 10 02:37:28 fdmi: C4 16 65 08 44 19 65 08 E4 87 39 08 04 17 65 08
2005 Mar 10 02:37:28 fdmi: 84 19 65 08 4C 8D 39 08 44 17 65 08 C4 19 65 08
2005 Mar 10 02:37:28 fdmi: B4 92 39 08 00 17 65 08 04 1A 65 08 1C 98 39 08
2005 Mar 10 02:37:28 fdmi: C4 17 65 08 44 1A 65 08 84 9D 39 08 04 18 65 08
2005 Mar 10 02:37:28 fdmi: 84 1A 65 08 EC A2 39 08 44 18 65 08 C4 1A 65 08
2005 Mar 10 02:37:28 fdmi: 54 A8 39 08 84 18 65 08 04 1B 65 08 BC AD 39 08
2005 Mar 10 02:37:28 fdmi: 00 00 00 02 00 00 0B B8 00 00 00 00 00 00 00 00
2005 Mar 10 02:37:28 fdmi: 00 00 00 00 00 00 00 00 00 00 00 00
2005 Mar 10 02:37:28 fdmi: Src: 0x00000601/27  Dst: 0x00000601/105  ID: 0x0069E217  Size:
140 [REQ] Opc: 7804 (MTS_OPC_FDMI_SNMP) RR: 0x0069E217  HA_SEQNO: 0x00000000  TS:
0x25218CC5A40E3  REJ:0  SYNC:0
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show fdmi	Displays the FDMI database information.

debug ficon

To enable debugging for the Fibre Connection (FICON) interface capabilities, use the **debug ficon** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug ficon {**all** | **bypass** {**acl** | **esa** | **file** | **pm** | **postcheck** | **precheck**} | **control-device** {**all** | **bypass ficon_mgr** | **demux** [**vsan** *vsan-id*] | **deque** | **error** | **event** [**vsan** *vsan-id*] | **ficon_mgr** | **ha** [**vsan** *vsan-id*] | **demux** [**vsan** *vsan-id*] | **sb3** {**error** | **flow**} **trace [detail]** [**vsan** *vsan-id*] | **warning** [**vsan** *vsan-id*]} | **error** | **event** | **file-trace** | **ha** | **max-port-number** *ports* | **pss-trace** | **stat** {**all** | **demux** [**vsan** *vsan-id*] | **deque** | **error** | **event** [**vsan** *vsan-id*] | **ha** [**vsan** *vsan-id*] | **trace** [**detail**] [**vsan** *vsan-id*] | **warning** [**vsan** *vsan-id*]} | **timer** | **trace**}

no debug ficon {**all** | **bypass** {**acl** | **esa** | **file** | **pm** | **postcheck** | **precheck**} | **control-device** {**all** | **bypass ficon_mgr** | **demux** [**vsan** *vsan-id*] | **deque** | **error** | **event** [**vsan** *vsan-id*] | **ficon_mgr** | **ha** [**vsan** *vsan-id*] | **demux** [**vsan** *vsan-id*] | **sb3** {**error** | **flow**} **trace [detail]** [**vsan** *vsan-id*] | **warning** [**vsan** *vsan-id*]} | **error** | **event** | **file-trace** | **ha** | **max-port-number** *port* | **pss-trace** | **stat** {**all** | **demux** [**vsan** *vsan-id*] | **deque** | **error** | **event** [**vsan** *vsan-id*] | **ha** [**vsan** *vsan-id*] | **trace** [**detail**] [**vsan** *vsan-id*] | **warning** [**vsan** *vsan-id*]} | **timer** | **trace**}

Syntax Description

all	Enables debugging for all FICON features.
bypass	Enables bypass flags for FICON error conditions.
acl	Bypasses ACL manager execution.
esa	Bypasses ESA execution.
file	Bypasses file operations execution.
pm	Bypasses port manager execution.
postcheck	Bypass es post check execution for VSAN enable.
precheck	Bypasses precheck execution for VSAN enable.
control-device	Enables the dump of FICON control devices.
all	Specifies all debug flags of FICON control device.
bypass ficon_mgr	Bypasses FICON Manager.
demux	Configures debugging of FICON control device message demux.
deque	Configures debugging of FICON control device message deque.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.
error	Configures debugging of FICON control device error.
event	Configures debugging of FICON control device FSM and Events.
ficon_mgr	Configures debugging of FICON manager control device.
ha	Configures debugging of FICON control device HA.
sb3	Configures debugging of SB3 library.
error	Enables debugging for FICON errors.
flow	
trace	Configures debugging of FICON control device trace.
detail	(Optional)
warning	Configures debugging of FICON control device warning.

error	Enables debugging for FICON errors.
event	Enables debugging for FICON events.
file-trace	Enables debugging of FICON file flow.
ha	Enables the debugging of HA synchronization messages.
max-port-number *ports*	Specifies maximum number of ports.
pss-trace	Enables debugging of FICON PSS flow.
stat	Enables debugging of FICON statistics.
all	Specifies all debug flags of FICON statistics.
demux	Specifies FICON statistics message demux.
deque	Specifies FICON statistics message deque.
error	Specifies FICON statistics errors.
event	Specifies FICON statistics FSM and events.
ha	Specifies FICON statistics HA.
trace	Specifies FICON statistics trace.
warning	Specifies FICON statistics warnings
timer	Enables debugging of FICON timer messages.
trace	Enables debugging of FICON flow.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.3(2)	This command was introduced.

Usage Guidelines

FICON must be enabled on the switch to use this command.

Examples

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

```
switch# debug ficon all
2005 Mar 10 02:38:58 ficon: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
2005 Mar 10 02:38:58 ficon: fu_fsm_execute_all: null fsm_event_list
2005 Mar 10 02:38:58 ficon: fu_fsm_engine_post_event_processing: mts msg
MTS_OPC_DEBUG_WRAP_MSG(msg_id 6943776) dropped
switch# undebug all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show ficon	Displays configured FICON information.

debug flogi

To enable debugging for the fabric login (FLOGI) feature, use the **debug flogi** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug flogi {**action** [**interface** *type number* | **vsan** *vsan-id*] | **all** | **bypass** {**acl** | **dm** | **dpvm** | **fcsp** | **lcp** | **npiv** | **ns** | **pl** | **pm** | **pmvc** | **rib**| **vsan_mgr** | **zs**} | **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**}

no debug flogi {**action** [**interface** *type number* | **vsan** *vsan-id*] | **all** | **bypass** {**acl** | **dm** | **dpvm** | **fcsp** | **lcp** | **npiv** | **ns** | **pl** | **pm** | **pmvc** | **rib**| **vsan_mgr** | **zs**} | **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

action	Enables all FLOGI debug features.
interface *type number*	(Optional) Restricts debugging to the specified interface.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.
all	Enables all FLOGI debug options.
bypass	Bypasses some components in FLOGI execution.
acl	Bypasses ACL execution.
dm	Bypasses domain manager execution.
dpvm	Bypasses DPVM execution.
fcsp	Bypasses FCSP execution.
lcp	Bypasses LCP execution.
npiv	Bypasses NPIV execution.
ns	Bypasses name server execution.
pl	Bypasses port lock execution.
pm	Bypasses port manager execution.
pmvc	Bypasses PM VSAN change execution.
rib	Bypasses RIB execution.
vsan_mgr	Bypasses VSAN manager execution.
zs	Bypasses zone server execution.
demux	Enables FLOGI demux
error	Enables debugging for FLOGI error conditions.
event	Enables debugging for FLOGI FSMs and events.
ha	Enables debugging for FLOGI high availability.
init	Enables debugging of FLOGI addition, deletion, and initialization.
timers	Enables debugging for FLOGI message timers.
trace	Enables debugging for FLOGI traces.
warning	Enables debugging for FLOGI warnings.

Defaults Disabled.

Command Modes EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show flogi database	Displays all the FLOGI sessions through all interfaces across all VSANs.

debug fm

To enable feature manager debugging, use the **debug fm** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fm {**error** | **flow**}

no debug fm {**error** | **flow**}

Syntax Description

error	Enables debugging for feature manager error conditions.
flow	Enables debugging for the feature manager flow.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug fm flow
switch# 2005 Mar 10 02:40:19 feature-mgr: fm_event_loop: ----------- EVENT START
2005 Mar 10 02:40:19 feature-mgr: fm_event_loop: received MTS message:
2005 Mar 10 02:40:19 feature-mgr: fm_event_loop: Src: 0x00000601/27  Dst: 0x00000601/121
ID: 0x006A0FC4  Size: 160 [REQ] Opc: 8922 (MTS_OPC_FM_CMI_GET_FEATURE_OP) RR: 0x006A0FC4
HA_SEQNO: 0x00000000  TS: 0x2524B48D52B53  REJ:0  SYNC:0
2005 Mar 10 02:40:19 feature-mgr: fm_handle_cmi_get_feature_op: Get feature (1) op request
2005 Mar 10 02:40:19 feature-mgr: fm_handle_cmi_get_feature_op: Reply to get feature ivr
op request: op 2, op_state 0, result 0x0 (success)
2005 Mar 10 02:40:19 feature-mgr: fm_event_loop: ----------- EVENT START
2005 Mar 10 02:40:19 feature-mgr: fm_event_loop: received MTS message:
2005 Mar 10 02:40:19 feature-mgr: fm_event_loop: Src: 0x00000601/27  Dst: 0x00000601/121
ID: 0x006A0FC6  Size: 160 [REQ] Opc: 8922 (MTS_OPC_FM_CMI_GET_FEATURE_OP) RR: 0x006A0FC6
HA_SEQNO: 0x00000000  TS: 0x2524B48EBF55D  REJ:0  SYNC:0
2005 Mar 10 02:40:19 feature-mgr: fm_handle_cmi_get_feature_op: Get feature (1) op request
2005 Mar 10 02:40:19 feature-mgr: fm_handle_cmi_get_feature_op: Reply to get feature ivr
op request: op 2, op_state 0, result 0x0 (success)
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug fspf

To enable debugging for the FSPF feature, use the **debug fspf** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug fspf {**all** [**interface** *type number*] [**vsan** *vsan-id*] | **database** [**interface** *type number*] [**vsan** *vsan-id* | **error** | **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** {**both** | **rx** [**node** *range* | **opcode** *range* | **sap** *range*] | **tx**} | **pkthdr** {**both** | **rx** [**numpkt** *range*] | **tx**}} | **retrans** [**interface** *type number*] [**vsan** *vsan-id*] | **route** | **timer**}

no debug fspf {**all** [**interface** *type number*] [**vsan** *vsan-id*] | **database** [**interface** *type number*] [**vsan** *vsan-id* | **error** | **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** {**both** | **rx** [**node** *range* | **opcode** *range* | **sap** *range*] | **tx**} | **pkthdr** {**both** | **rx** [**numpkt** *range*] | **tx**}} | **retrans** [**interface** *type number*] [**vsan** *vsan-id*] | **route** | **timer**}

Syntax Description

all	Enables debugging for all FSPF features.
interface *type number*	(Optional) Restricts debugging to the specified interface.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.
database	Enables debugging for the FSPF database.
error	Enables debugging for FSPF error conditions.
event	Enables debugging for FSPF events.
fc	Enables debugging of Fibre Channel packets and headers.
pkt	Enables debugging for FCC tx/rx FCC packets.
pkthdr	Enables debugging for FCC tx/rx FCC headers.
both	Specifies debugging in both the transmit and receive directions.
tx	Specifies debugging in the transmit direction,
rx	Specifies debugging in the receive direction.
flood	Enables debugging for FSPF flodding events.
ha	Enables debugging for FSPF high availability.
mts	Enables debugging for FSPF tx/rx MTS events.
node *range*	(Optional) Specifies the node for the packets in the receive direction. The integer range is from 1 to 4096.
opcode *range*	(Optional) Specifies the opcode for the packets in the receive direction. The integer range is from 1 to 4096.
sap *range*	(Optional) Specifies the sap for the packets in the receive direction. the integer range is from 1 to 4096.
numpkt *range*	(Optional) Specifies the number of required packets
retrans	Enables debugging for FSPF retransmits.
route	Enables debugging for FSPF route computation.
timer	Enables debugging for FSPF timers.

Defaults Disabled.

Command Modes EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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, enter **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
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show fspf	Displays global FSPF information.

debug hardware arbiter

To configure debugging for the hardware arbiter driver, use the **debug hardware arbiter** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug hardware arbiter {**error** | **flow**} [**group** *number*]}

no debug hardware arbiter {**error** | **flow**} [**group** *number*]}

Syntax Description

error	Enables debugging for hardware arbiter kernel errors.
flow	Enables debugging for hardware arbiter kernel flow.
group *number*	(Optional) Restricts debugging to the specified group. The range is 0 to 17.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show hardware	Displays switch hardware inventory details.

debug idehsd

To enable IDE hot swap handler debugging, use the **debug idehsd** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug idehsd {**cmd dbglevel** [*debug-level*] | **error** | **flow**}

no debug idehsd {**cmd dbglevel** [*debug-level*] | **error** | **flow**}

Syntax Description

cmd dbglevel	Enables debugging for the IDE hot swap handler.
debug-level	(Optional) Specifies the debug level (0 to 8).
error	Enables debugging for IDE hot swap handler error conditions.
flow	Enables debugging for IDE hot swap handler flow.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug ike

To enable debugging for the IKE protocol, use the **debug ike** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug ike {**all** | **error** | **event** | **message** | **mts** | **protocol** | **verbose** | **warning**}

no debug ike {**all** | **error** | **event** | **message** | **mts** | **protocol** | **verbose** | **warning**}

Syntax Description

all	Enables all of the debugging flags for IKE.
error	Enables debugging for IKE errors.
event	Enables debugging for IKE event generation.
message	Enables debugging for IKE messages.
mts	Enables debugging for MTS-related IKE activity.
protocol	Enables debugging for IKE protocol-related handling.
verbose	Enables verbose debugging for IKE protocol-related handling.
warning	Enables debugging for IKE warnings.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
2.0(x)	This command was introduced.

Usage Guidelines

To use this command, IKE must be enabled using the **crypto ike enable** command.

Examples

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

```
switch# debug ike all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show crypto ike domain ipsec	Displays IKE protocol information.

debug ilc_helper

To enable ILC helper debugging, use the **debug ilc_helper** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug ilc_helper {**all** | **errors** | **events** | **info**}

no debug ilc_helper {**all** | **errors** | **events** | **info**}

Syntax Description

all	Enables debugging for all ILC helper features.
errors	Enables debugging for ILC helper error conditions.
events	Enables debugging for the ILC helper events.
info	Enables debugging for ILC helper information.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug ilc_helper all
For Application :125, sdwrap:mts_send : Broken pipe
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug ipacl

To enable IP access control list (ACL) debugging, use the **debug ipacl** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug ipacl {**all** | **error** | **event** | **trace**}

no debug ipacl {**all** | **error** | **event** | **trace**}

Syntax Description

all	Enables debugging for all IP ACL features.
error	Enables debugging for IP ACL error conditions.
event	Enables debugging for the IP ACL events.
trace	Enables debugging for IP ACL trace.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug ipacl all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show ip access-list	Displays the IP access control lists that are currently active.

debug ipconf

To enable IP configuration debugging, use the **debug ipconf** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug ipconf {all | errors | events | info | pss}

no debug ipconf {all | errors | events | info | pss}

Syntax Description

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

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug ipconf all
switch# 2005 Mar 10 02:45:30 ipconf: Received MTS message
2005 Mar 10 02:45:30 ipconf: MTS message received opcode 862 source 0x00000601/27
2005 Mar 10 02:45:30 ipconf: Getting ip addresses on interface 5000000
2005 Mar 10 02:45:30 ipconf: Received MTS message
2005 Mar 10 02:45:30 ipconf: MTS message received opcode 862 source 0x00000601/27
2005 Mar 10 02:45:30 ipconf: Getting ip addresses on interface 5000000
2005 Mar 10 02:45:30 ipconf: Received MTS message
2005 Mar 10 02:45:30 ipconf: MTS message received opcode 862 source 0x00000601/27
2005 Mar 10 02:45:30 ipconf: Getting ip addresses on interface 5000000
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug ipfc

To enable IP over Fibre Channel (IPFC) debugging, use the **debug ipfc** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug ipfc {**all** | **errors** | **events** | **info** | **kernel** {**errors** | **events**}}

Syntax Description

all	Enables debugging for all IPFC features.
errors	Enables debugging for IPFC error conditions.
events	Enables debugging for IPFC tx/rx MTS events.
info	Enables debugging for IPFC information.
kernel	Enables debugging for IPFC kernel operations.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug ips

To enable debugging for the IP Storage Services (IPS) module, use the **debug ips** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug ips {**acl** {**flow** | **flow-detail**} | **all** | **demux** | **error** | **flow** {**ethernet** | **fcip**} | **fsm** | **ha** | **init** | **iscsi** {**config** | **config-detail** | **flow** | **flow-detail** | **msgs**} | **islb** {**cfs** {**config** | **config-detail** | **error** | **flow** | **flow-detail**} | **config** | **config-detail** | **flow** | **flow-detail** | **vrrp** {**error** | **flow** | **flow-detail**}} | **isns** {**config** | **config-detail** | **error** | **flow** | **flow-detail** | **msgs** | **packet**} | **show_all** | **upgrade**}

no debug ips {**acl** {**flow** | **flow-detail**} | **all** | **demux** | **error** | **flow** {**ethernet** | **fcip**} | **fsm** | **ha** | **init** | **iscsi** {**config** | **config-detail** | **flow** | **flow-detail** | **msgs**} | **islb** {**cfs** {**config** | **config-detail** | **error** | **flow** | **flow-detail**} | **config** | **config-detail** | **flow** | **flow-detail** | **vrrp** {**error** | **flow** | **flow-detail**}} | **isns** {**config** | **config-detail** | **error** | **flow** | **flow-detail** | **msgs** | **packet**} | **show_all** | **upgrade**}

Syntax Description

acl	Enables debugging for ACLs.
flow	Enables debugging for the IPS flow.
flow-detail	Enables detailed debugging for the IPS flow.
all	Enables all IPS debug options.
demux	Enables debugging for IPS demux.
error	Enables debugging for IPS error conditions.
ethernet	Restricts debugging to the Ethernet flow.
fcip	Restricts debugging to the FCIP flow.
fsm	Enables debugging for IPS FSM and events.
ha	Enables debugging for IPS high availability.
init	Enables debugging of IPS addition, deletion, and initialization.
iscsi	Enables debugging of iSCSI.
config	Enables debugging of the iSCSI configuration.
config-detail	Enables detailed debugging of the iSCSI configuration.
msgs	Enables debugging of the iSCSI messages received and responded.
islb	Enables debugging of iSLB.
cfs	Enables debugging of iSLB CFS.
error	Enables debugging of iSLB CFS error conditions.
flow	Enables debugging for the iSLB CFS flow.
flow-detail	Enables detailed debugging for the iSLB CFS flow.
vrrp	Enables debugging of iSLB VRRP.
error	Enables debugging of iSNS error conditions.
msgs	Enables debugging of the iSNS messages received and responded.
packet	Enables debugging of an iSNS packet.
show_all	Enables all debugging IPS manager flags.
upgrade	Enables debugging for upgrade.

Defaults Disabled.

Command Modes EXEC mode.

Command History

Release	Modification
1.1(1)	This command was introduced.
3.0(1)	Added the **iSLB** and **iSNS** options.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show ips stats	Displays IP storage statistics.
show ips status	Displays the IP storage status.

debug ipsec

To enable debugging for IPsec, use the **debug ipsec** command in EXEC mode.To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug ipsec {**all** | **bypass ficon_mgr** | **config** | **config-detail** | **demux** | **deque** | **error** | **event** | **flow** | **flow-detail** | **ha** | **trace** [**detail**] | **warning**}

no debug ipsec {**all** | **bypass ficon_mgr** | **config** | **config-detail** | **demux** | **deque** | **error** | **event** | **flow** | **flow-detail** | **ha** | **trace** [**detail**] | **warning**}

Syntax Description

all	Enables all IPsec debugging.
bypass ficon_mgr	Bypasses the FICON manager.
config	Enables debugging for IPsec configuration.
config-detail	Enables debugging for detailed IPsec configuration.
demux	Enables debugging for IPsec message demux.
deque	Enables debugging for IPsec message dequeue.
error	Enables debugging for IPsec errors.
event	Enables debugging for IPsec FSM and events.
flow	Enables debugging for IPsec flow.
flow-detail	Enables debugging for detailed IPsec flow.
ha	Enables debugging for IPsec high availability.
trace	Enables debugging for IPsec trace.
detail	(Optional) Specifies detailed trace.
warning	Enables debugging for IPsec warning.

Defaults

None.

Command Modes

EXEC mode.

Command History

Release	Modification
2.0(x)	This command was introduced.

Usage Guidelines

To use this command, IPsec must be enabled using the **crypto ipsec enable** command.

Examples

The following example displays the system output when the **debug ipsec config** command is issued.

```
switch# debug ipsec config
```

Related Commands

Command	Description
crypto ipsec enable	Enables IPsec.
no debug all	Disables all debugging.

debug isns

To enable debugging for Internet storage name services (iSNS), use the **debug isns** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug isns {**all** | **bypass ficon_mgr** | **cloud** | **db** | **deque** | **error** | **event** [**vsan** *vsan-id*] | **fabric distribute** | **ha** [**vsan** *vsan-id*] | **prot** | **trace** [**detail**] | **warning** [**vsan** *vsan-id*]}

no debug isns {**all** | **bypass ficon_mgr** | **cloud** | **db** | **deque** | **error** | **event** [**vsan** *vsan-id*] | **fabric distribute** | **ha** [**vsan** *vsan-id*] | **prot** | **trace** [**detail**] | **warning** [**vsan** *vsan-id*]}

Syntax Description

all	Enables all iSNS debugging.
bypass ficon_mgr	Enables bypassing FICON manager execution.
cloud	Enables debugging for iSNS cloud discovery.
db	Enables debugging for iSNS database.
deque	Enables debugging for iSNS message dequeue.
error	Enables debugging for iSNS error.
event	Enables debugging for iSNS event.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN ID. The range is 1 to 4093.
fabric distribute	Enables debugging for iSNS fabric distribution.
ha	Enables debugging for iSNS high availability.
prot	Enables debugging for iSNS protocol.
trace	Enables debugging for iSNS trace.
detail	(Optional) Enables detailed iSNS trace.
warning	Enables debugging for iSNS warning.

Defaults

None.

Command Modes

EXEC mode.

Command History

Release	Modification
2.0(x)	This command was introduced.

Usage Guidelines

To use this command, iSNS must be enabled using the **isns-server enable** command.

Examples

The following example displays the system output when the **debug isns error** command is issued.

```
switch# debug isns error
```

Related Commands

Command	Description
isns-server enable	Enables the iSNS server.
no debug all	Disables all debugging.
show isns	Displays iSNS information.

debug ivr

To enable debugging for inter-VSAN routing (IVR), use the **debug ivr** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug {**all** | **demux** | **dep** | **dep-detail** | **dequeue** | **drav-fsm** | **drav-fsm-detail** | **errors** | **fcid-rewrite** | **fcid-rewrite-detail** | **ficon** | **ficon-detail** | **ha** | **pnat** | **pv** | **pv-detail** | **state-machine** [**vsan** *vsan-id*] | **test** | **trace** | **trace-detail** | **tu-fsm** | **tu-fsm-detai***l* | **zone-distrib-errors** | **zone-distrib-events** | **zone-fsm** | **zone-fsm-detail**}

no debug {**all** | **demux** | **dep** | **dep-detail** | **dequeue** | **drav-fsm** | **drav-fsm-detail** | **errors** | **fcid-rewrite** | **fcid-rewrite-detail** | **ficon** | **ficon-detail** | **ha** | **pnat** | **pv** | **pv-detail** | **state-machine** [**vsan** *vsan-id*] | **test** | **trace** | **trace-detail** | **tu-fsm** | **tu-fsm-detai***l* | **zone-distrib-errors** | **zone-distrib-events** | **zone-fsm** | **zone-fsm-detail**}

Syntax Description

all	Enables all filters for IVR debugging.
demux	Enables debugging of IVR event demultiplexing.
dep	Enables debugging of IVR DEP.
dep-detail	Enables debugging of IVR DEP detail.
dequeue	Enables debugging of IVR event dequeue.
drav-fsm	Enables debugging of IVR DRAV finite state machine (FSM).
drav-fsm-detail	Enables debugging of IVR DRAV FSM detail.
errors	Enables debugging for IVR errors.
fcid-rewrite	Enables debugging of IVR FC ID rewrite.
fcid-rewrite-detail	Enables debugging of IVR FC ID rewrite detail.
ficon	Enables debugging of IVR FICON.
ficon-detail	Enables debugging of IVR FICON detail.
ha	Enables debugging of IVR high-availability.
pnat	Enables debugging of IVR payload Network Address Translation (NAT).
pv	Enables debugging of IVR PV state machine.
pv-detail	Enables debugging of IVR PV state machine detail.
state-machine	Enables debugging of FSM.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.
test	Enables debugging of IVR test features.
trace	Enables debugging of IVR trace.
trace-detail	Enables debugging of IVR detail trace.
tu-fsm	Enables debugging of IVR TU FSM.
tu-fsm-detail	Enables debugging of IVR TU FSM detail.
zone-distrib-errors	Enables debugging of IVR zone distribution errors.
zone-distrib-events	Enables debugging of IVR zone distribution events.
zone-fsm	Enables debugging of IVR zone FSM.
zone-fsm-detail	Enables debugging of IVR zone FSM detail.

Defaults Disabled.

Command Modes EXEC mode.

Command History

Release	Modification
2.1(1)	This command was introduced.
3.0(1)	• Added the **ficon** and **ficon-detail** options.

Usage Guidelines None.

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

```
switch# debug ivr all
2005 Mar 10 01:27:27 ivr: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
2005 Mar 10 01:27:27 ivr: fu_fsm_execute_all: null fsm_event_list
2005 Mar 10 01:27:27 ivr: fu_fsm_engine_post_event_processing: mts msg
MTS_OPC_DEBUG_WRAP_MSG(msg_id 6774251) dropped
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show ivr	Displays IVR configurations.

debug klm

To enable kernel loadable module parameter debugging, use the **debug klm** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug klm {**fc2** {**cpuhog** *seconds* | **flag** *flags*} | **scsi-target** {**driver** | **error** [**vsan** *vsan-id*] [**fcid** *fc-id*] | **flag** *flags* | **flow** [**vsan** *vsan-id*] [**fcid** *fc-id*] | **snmp** | **syscall**} | **sdip** {**all** | **error** | **flow** | **warning**}}

no debug klm {**fc2** {**cpuhog** *seconds* | **flag** *flags*} | **scsi-target** {**driver** | **error** [**vsan** *vsan-id*] [**fcid** *fc-id*] | **flag** *flags* | **flow** [**vsan** *vsan-id*] [**fcid** *fc-id*] | **snmp** | **syscall**} | **sdip** {**all** | **error** | **flow** | **warning**}}

Syntax Description

fc2	Enables debugging for FC2 driver debug parameters.
cpuhog *seconds*	Specifies the FC2 CPU hog value. The ranges is 0 to 10000 seconds.
flag *flags*	Specifies the flag values. The ranges is 0x0 to 0xffffffff.
scsi-target	Enables debugging for the SCSI target driver.
driver	Enables debugging for SCSI target driver flags.
error	Enables debugging for driver error conditions.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.
fcid *fc-id*	(Optional) Restricts debugging to the specified FCID interface.
flow	Enables debugging for SCSI target flow.
snmp	Enables debugging for SCSI target SNMP requests.
syscall	Enables debugging for SCSI target system call request.
sdip	Enables debugging for the SDIP driver.
all	Enables debugging for the SCSI target driver.
flow	Enables debugging for driver flow.
warning	Enables debugging for driver warnings.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug klm scsi-target driver
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug license

To enable licensing debugging, use the **debug license** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug license {**all** | **errors** | **event s** | **mts**}

no debug license {**all** | **errors** | **events** | **mts**}

Syntax Description

all	Enables debugging for all licensing features.
errors	Enables debugging for licensing error conditions.
events	Enables debugging for the licensing events.
mts	Enables debugging for Tx/Rx packets of MTS.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug license all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show license	Displays license information.

debug logfile

To direct the output of the debug commands to a specified file, use the **debug logfile** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug logfile *filename* [**size** *bytes*]

Syntax Description

filename	Assigns the name of the log file. Maximum length is 80 characters.
size *bytes*	(Optional) Specifies the logfile size in bytes. The range is 4096 to 4194304.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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.

When you use the **debug logfile** command to create a log file, the file is automatically created in the log: directory on the supervisor module unless you specify a different path.

For example, when you use the **debug logfile** command to create a log file named captureDebug, you must enter the **dir log://sup-local/?** command to find the log file you created. This example shows you how to find the log file created:

```
switch# debug logfile captureDebug
switch#  dir log://sup-local/?
  log:                        Enter URL "log:[//<module-number>]/<filename>"
  log://sup-local/dmesg
  log://sup-local/messages
→ log://sup-local/captureDebug

switch#  dir log://sup-local/
```

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show logging	Displays the current message logging configuration.

debug mcast

To enable debugging for multicast definitions, use the **debug mcast** command in EXEC mode. To disable a **debug** command, use the **no** form of the 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** | **rx** [**node** *range* | **opcode** *range* | **sap** *range*] | **tx**} | **pkthdr** {**both** | **rx** [**numpkt** *range*] | **tx**}} | **trace** [**vsan** *vsan-id*] [**interface fc** *slot*/*port*]]

no debug mcast {**all** | **error** [**vsan** *vsan-id*] [**interface fc** *slot*/*port*] | **event** [**vsan** *vsan-id*] [**interface fc** *slot*/*port*] | **mts** {**pkt** {**both** | **rx** [**node** *range* | **opcode** *range* | **sap** *range*] | **tx**} | **pkthdr** {**both** | **rx** [**numpkt** *range*] | **tx**}} | **trace** [**vsan** *vsan-id*] [**interface fc** *slot*/*port*]]

Syntax Description

all	Enables debugging for all multicast definitions.
error	Enables debugging for multicast errors.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.
interface fc *slot/port*	(Optional) Restricts debugging to the specified interface.
event	Enables debugging for multicast events.
mts	Enables debugging for multicast tx/rx MTS events.
pkt	Specifies debugging of packets.
both	Specifies debugging in both the transmit and receive direction.
rx	Specifies debugging in the receive direction.
node *range*	Specifies the node for the packets in the receive direction. The integer range is from 1 to 4096.
opcode *range*	Specifies the opcode for the packets in the receive direction. The integer range is from 1 to 4096.
sap *range*	Specifies the sap for the packets in the receive direction.The integer range is from 1 to 4096.
tx	Specifies debugging in the transmit direction,
pkthdr	Specifies debugging of headers.
numpkt	Specifies the number of required packets.
trace	Enables debugging for multicast traces.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug mcast all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show mcast	Displays multicast information.

debug mip

To enable debugging for multiple IP (MIP) kernel drivers, use the **debug mip** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug mip {**errors** | **events**}

no debug mip {**errors** | **events**}

Syntax Description

errors	Enables debugging for MIP error conditions.
events	Enables debugging for MIP events.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug mip errors
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug module

To enable debugging for switching or service modules, use the **debug module** command in EXEC mode. To disable a **debug** command, use the **no** form of the 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*]}

no debug module {**all** | **error** [**module** *slot*] | **event** | **ha** | **no-powerdown** | **trace** [**module** *slot*]}

Syntax Description

all	Enables debugging for all module features.
error	Enables debugging for module error conditions.
module *slot*	(Optional) Restricts debugging to the specified module.
event	Enables debugging for module events.
ha	Enables debugging for a module's high availability features.
no-powerdown	Disables the power cycle feature for the module.
trace	Enables debugging for a module's trace flows.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug module all
2005 Mar 10 02:51:01 module: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
2005 Mar 10 02:51:01 module: fu_fsm_execute_all: null fsm_event_list
2005 Mar 10 02:51:01 module: fu_fsm_engine_post_event_processing: mts msg
MTS_OPC_DEBUG_WRAP_MSG(msg_id 6986564) dropped
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show module	Displays the status of a module.

debug ntp

To enable debugging for the Network Time Protocol (NTP) module, use the **debug ntp** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug ntp {**errors** | **info**}

no debug ntp {**errors** | **info**}

Syntax Description

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

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug ntp info
2005 Mar 10 03:00:42 ntp: Dropping msg_ref with rr_token [7002722]
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show ntp	Displays the configured NTP server and peer associations.

debug npv

To enable debugging N Port Virtualization (NPV) configuration on the switch, use the **debug npv** command.

debug npv

Syntax Description

This command has no arguments or keywords.

Defaults

None.

Command Modes

EXEC mode.

Command History

Release	Modification
3.2(1)	This command was introduced.

Usage Guidelines

None.

Examples

The following example shows all NPV debug commands configured on the switch:

```
switch# show debug npv
N_port Virtualizer:
 FC Receive Packets debugging is on
 FC Transmit Packets debugging is on
 FC Receive Packet header debugging is on
 FC Transmit Packet header debugging is on
 MTS Receive Packets debugging is on
 MTS Transmit Packets debugging is on
 MTS Receive Packet header/payload debugging is on
 MTS Transmit Packet header/payload debugging is on
 High Availability debugging is on
 FSM Transitions debugging is on
 Error debugging is on
 Warning debugging is on
 Trace debugging is on
 Trace Detail debugging is on
 Demux debugging is on
 Dequeue debugging is on
 Packets debugging is on
 Database debugging is on
 Timers debugging is on
 External Interface FSM Events debugging is on
 External Interface FSM Errors debugging is on
 External Interface FSM Trace debugging is on
 FLOGI FSM Events debugging is on
 FLOGI FSM Errors debugging is on
 FLOGI FSM Trace debugging is on
 Server Interface FSM Events debugging is on
```

```
Server Interface FSM Errors debugging is on
Server Interface FSM Trace debugging is on
Events debugging is on
```

Related Commands

Command	Description
show debug npv	Displays the NPV debug commands configured on the switch.

debug obfl

To enable debugging for Onboard Failure Logging (OBFL), use the **debug obfl** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug obfl {**error** | **trace**}

no debug obfl {**error** | **trace**}

Syntax Description

error	Enables debugging for OBFL error conditions.
trace	Enables debugging for OBFL events.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
3.0(1)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug obfl error
2006 Jan 23 21:30:59.573503 obfl: obfl_process_mts_msgs(): OBFL received mts mes
sage: opc:182
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show logging onboard	Displays OBFL information.

debug platform

To enable debugging for the platform manager, use the **debug platform** command in EXEC mode. To disable a **debug** command, use the **no** form of the 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**} {**tx** | **rx**} | **nopowerdown** | **supervisor-reset**}

no debug platform {**all** [**fc_id** *fc-id*] | **error** [**module** *slot*] | **flow** [**module** *slot*] | **fsm** | **ha** | **hitless** | **mts** {**pkt** | **pkthdr**} {**tx** | **rx**} | **nopowerdown** | **supervisor-reset**}

Syntax Description

all	Enables debugging for all platform features.
fcid *fc-id*	(Optional) Restricts debugging to the specified FC ID module number. The range is 0 to 2147483647.
error	Enables debugging for platform-related error conditions.
module *slot*	Restricts debugging to the specified module.
flow	Enables debugging for platform-related flows.
fsm	Enables debugging for platform-related FSMs.
ha	Enables debugging for platform-related high availability.
hitless	Enables the platform loading feature while the switch is in hitless mode.
mts	Enables debugging for platform-related tx/rx MTS events.
pkt	Enables debugging of packets.
pkthdr	Enables debugging of headers.
tx	Enables debugging in the transmit direction.
rx	Enables debugging in the receive direction.
nopowerdown	Enables powering down modules
supervisor-reset	Resets the local supervisor.
pkt	Enables debugging of packets.
pkthdr	Enables debugging of headers.
tx	Enables debugging in the transmit direction.
rx	Enables debugging in the receive direction.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug platform all
2005 Mar 10 03:01:56 platform: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
2005 Mar 10 03:01:56 platform: fu_fsm_execute_all: null fsm_event_list
2005 Mar 10 03:01:56 platform: fu_fsm_engine_post_event_processing: mts msg
MTS_OPC_DEBUG_WRAP_MSG(msg_id 7004045) dropped
v-185# 2005 Mar 10 03:01:56 platform: env_chg_none: ps 0 old 1 new 1
2005 Mar 10 03:01:57 platform: env_chg_none: ps 0 old 1 new 1
2005 Mar 10 03:01:58 platform: env_chg_none: ps 0 old 1 new 1
v-185# debug platform all
2005 Mar 10 03:01:59 platform: fu_priority_select:  - setting fd[7] for select call
2005 Mar 10 03:01:59 platform: fu_priority_select_select_queue: round credit(5)
2005 Mar 10 03:01:59 platform:     curr_q - FU_PSEL_Q_CAT_CQ, usr_q_info(0), priority(1),
credit(0), empty
2005 Mar 10 03:01:59 platform: fu_priority_select: returning FU_PSEL_Q_CAT_FD queue,
fd(7), usr_q_info(1)
2005 Mar 10 03:01:59 platform: fu_fsm_engine: line[2139]
.
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug plog

To enable debugging of persistent logging (PLOG), use the **debug plog** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug plog {**error | trace**}

no debug plog {**error | trace**}

Syntax Description

error	Enables debugging of PLOG error conditions.
trace	Enables debugging of PLOG events.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
3.0(1)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug plog
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug port

To enable debugging for ports, use the **debug port** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug port {**all** | **bypass** {**acl_manager** | **domain_manager** | **fcsp** | **ficon** | **fport_server** | **lcp** | **loopback_diag** | **port_channel_mgr** | **port_lock** | **qos_mgr** | **span** | **switch_wwn** | **vsan_mgr** | **wwn_mgr** |**xbar_mgr** | **zone_server**} | **error** | **event** [**interface** *type number* | **module** *slot*] | **ha** [**interface** *type number* | **module** *slot*] | **trace** [**interface** *type number* | **module** *slot*]}

no debug port {**all** | **bypass** {**acl_manager** | **domain_manager** | **fcsp** | **ficon** | **fport_server** | **lcp** | **loopback_diag** | **port_channel_mgr** | **port_lock** | **qos_mgr** | **span** | **switch_wwn** | **vsan_mgr** | **wwn_mgr** | **xbar_mgr** | **zone_server**} | **error** | **event** [**interface** *type number* | **module** *slot*] | **ha** [**interface** *type number* | **module** *slot*] | **trace** [**interface** *type number* | **module** *slot*]}

Syntax Description

all	Enables all port debug options.
bypass	Bypasses some components in port execution.
acl_manager	Bypasses ACL manager execution.
domain_manager	Bypasses domain manager execution.
fcsp	Bypasses FCSP execution.
ficon	Bypasses FICON execution.
fport_server	Bypasses FPort server execution.
lcp	Bypasses LCP execution.
loopback_diag	Bypasses loopback diagnostics execution.
port_channel_mgr	Bypasses PortChannel manager execution.
port_lock	Bypasses port lock execution.
qos_mgr	Bypasses QOS manager execution.
span	Bypasses SPAN execution.
switch_wwn	Bypasses using switch WWN and uses VSAN WWN in ELP.
vsan_mgr	Bypasses VSAN manager execution.
wwn_mgr	Bypasses WWN manager execution.
xbar_mgr	Bypasses XBAR manager execution.
error	Enables debugging for port error conditions.
event	Enables debugging for port FSMs and events.
interface *type number*	(Optional) Restricts debugging to the specified interface.
module *slot*	(Optional) Restricts debugging to the specified module.
ha	Enables debugging for port high availability.
trace	Enables debugging for port traces.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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)
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug port-channel

To enable debugging for PortChannels, use the **debug port-channel** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug port-channel {**all** | **error** | **event** | **ha** | **trace** | **warning**}

no debug port-channel {**all** | **error** | **event** | **ha** | **trace** | **warning**}

Syntax Description

all	Enables all PortChannel debug options.
error	Enables debugging for PortChannel error conditions.
event	Enables debugging for PortChannel FSMs and events.
ha	Enables debugging for PortChannel high availability.
trace	Enables debugging for PortChannel traces.
warning	Enables debugging for PortChannel warning.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
2005 Mar 10 03:03:26 port_channel: fu_fsm_execute_all: match_msg_id(0),
log_already_open(0)
2005 Mar 10 03:03:26 port_channel: fu_fsm_execute_all: null fsm_event_list
2005 Mar 10 03:03:26 port_channel: fu_fsm_engine_post_event_processing: mts msg
MTS_OPC_DEBUG_WRAP_MSG(msg_id 7005958) dropped
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show port-channel	Displays information about existing PortChannel configurations.

debug port-resources

To enable debugging for a port resources module, use the **debug port-resources** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug port-channel {**all** | **demux** | **deque** | **error** | **event** | **ha** | **mts** | **trace** | **warning**}

no debug port-channel {**all** | **demux** | **deque** | **error** | **event** | **ha** | **mts** | **trace** | **warning**}

Syntax Description

all	Enables all port resources debug options.
demux	Enables debugging of port resources messages.
deque	Enables debugging of port resources message dequeues.
error	Enables debugging for port resources error conditions.
event	Enables debugging for port resources FSMs and events.
ha	Enables debugging for port resources high availability.
mts	Enables debugging for port resources message MTS events.
trace	Enables debugging for port resources traces.
warning	Enables debugging for port resources warning.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
3.0(1)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug port-resources demux** command is issued:

```
switch# debug port-resources demux vsan 2
2006 Jan 19 22:10:59.244892 port-resources: fu_priority_select:  - setting fd[5]
 for select call
2006 Jan 19 22:10:59.244985 port-resources: fu_priority_select_select_queue: rou
nd credit(12)
2006 Jan 19 22:10:59.245018 port-resources:     curr_q - FU_PSEL_Q_CAT_CQ, usr_q
_info(2), priority(7), credit(6), empty
2006 Jan 19 22:10:59.245051 port-resources: fu_priority_select: returning FU_PSE
L_Q_CAT_MTS queue, fd(5), usr_q_info(1)
2006 Jan 19 22:10:59.245168 port-resources: prm_get_data_from_queue(664): dequeued mts msg
(128136), MTS_OPC_DEBUG_WRAP_MSG
```

```
2006 Jan 19 22:10:59.245205 port-resources: fu_fsm_engine: line[2205]
2006 Jan 19 22:10:59.245248 port-resources: prm_demux: ev[0]
ips-hac2# 2006 Jan 19 22:10:59.246440 port-resources: fu_fsm_execute_all: match_
msg_id(0), log_already_open(0)
2006 Jan 19 22:10:59.246507 port-resources: fu_fsm_execute_all: null fsm_event_list
2006 Jan 19 22:10:59.246578 port-resources: fu_fsm_engine_post_event_processing:
 mts msg MTS_OPC_DEBUG_WRAP_MSG(msg_id 128136) dropped
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show port-resources module	Displays information about port resources in a Generation 2 module.

debug qos

To enable debugging for quality of service (QoS), use the **debug qos** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug qos {**all** [**interface fc** *slot*/*port*] | **detail** | **errors supervisor** | **flow** | **trace**}

no debug qos {**all** [**interface fc** *slot*/*port*] | **detail** | **errors supervisor** | **flow** | **trace**}

Syntax Description

all	Enables all QoS debug options.
interface fc *slot/port*	(Optional) Restricts debugging to the specified interface.
detail	Enables all QoS debug output.
errors supervisor	Enables debugging for supervisor QoS error conditions.
flow	Enables flow-level QoS debug options.
trace	Enables debugging for QoS traces.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug qos all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show qos	Displays the current QoS settings along with a the number of frames marked high priority.

debug radius

To enable debugging for boot variables, use the **debug radius** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug radius {**aaa-request** | **aaa-request-lowlevel** | **all** | **config** | **config-lowlevel** | **server-monitor** | **server-monitor-errors**}

no debug radius {**aaa-request** | **aaa-request-lowlevel** | **all** | **config** | **config-lowlevel** | **server-monitor** | **server-monitor-errors**}

Syntax Description

aaa-request	Enables RADIUS AAA request debug.
aaa-request-lowlevel	Enables RADIUS AAA request low-level debugging.
all	Enables Enable all the debug flags.
config	Enables RADIUS configuration debugging.
config-lowlevel	Enables RADIUS configuring low-level debugging.
server-monitor	Enables RADIUS server monitoring.
server-monitor-errors	Enables RADIUS server monitor errors.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.3(1)	This command was introduced.
3.0(1)	Added the **server-monitor** and **server-monitor-errors** options.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug radius config-lowlevel** command is issued:

```
switch# debug radius config-lowlevel
Nov 20 06:36:42 radius: radius_new_debug_conf_open: entering...
Nov 20 06:36:42 radius: radius_new_conf_close: entering...
Nov 20 06:36:42 radius: radius_new_conf_close: returning 0
Nov 20 06:36:42 radius: radius_new_enable_info_config: entering for Radius Daemon debug
Nov 20 06:36:42 radius: radius_new_debug_conf_open: entering...
Nov 20 06:36:42 radius: radius_new_debug_conf_open: exiting
Nov 20 06:36:42 radius: radius_new_enable_info_config: SET_REQ for Radius Daemon debug
with 1
Nov 20 06:36:42 radius: radius_new_enable_info_config: SET_REQ done for Radius Daemon
debug with 1
```

```
Nov 20 06:36:42 radius: radius_new_enable_info_config: got back the return value of
configuration operation:success
Nov 20 06:36:42 radius: radius_new_debug_conf_close: entering...
Nov 20 06:36:42 radius: radius_new_debug_conf_close: returning 0
Nov 20 06:36:42 radius: radius_new_enable_info_config: exiting for Radius Daemon debug
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show radius	Displays the RADIUS Cisco Fabric Services (CFS) distribution status and other details.

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. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug rd-reg [*device-name* | *register address*]

Syntax Description

device-name	(Optional) Specifies the device name for the required device.
register address	(Optional) Specifies the register address for the required device.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
switch#
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug rdl errors

To enable debugging for RDL errors, use the **debug rdl errors** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug rdl errors

no debug rdl errors

Syntax Description

This command has no other arguments or keywords.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug rdl errors
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug rib

To enable debugging for the routing information base (RIB) feature, use the **debug rib** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug rib {**all** | **detail** | **error** | **event** | **liod_error** | **liod_event** | **liod_trace** | **trace**}

no debug rib {**all** | **detail** | **error** | **event** | **liod_error** | **liod_event** | **liod_trace** | **trace**}

Syntax Description

all	Enables debugging for all RIB features.
detail	Enables detailed debugging for all RIB features.
error	Enables debugging for RIB errors.
event	Enables debugging for RIB events.
liod_error	Enables debugging for lossless in-order delivery (LIOD) errors.
liod_event	Enables debugging for LIOD errors.
liod_trace	Enables debugging for LIOD trace events.
trace	Enables debugging for trace events.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.
3.0(1)	Added the **liod_error**, **liod_event**, and **liod_trace** options.

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 shows the **debug rib error** command:

```
switch# debug rib error
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug rlir

To enable Registered Link Incident Report (RLIR) debugging, use the **debug rlir** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug rlir {**all** | **errors** | **events** | **mts-errors** | **mts-events**}

no debug rlir {**all** | **errors** | **events** | **mts-errors** | **mts-events**}

Syntax Description

all	Enables debugging for all RLIR features.
errors	Enables debugging for RLIR error conditions.
events	Enables debugging for the RLIR events.
mts-errors	Enables debugging for MTS error conditions.
mts-events	Enables debugging for MTS events.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug rlir all
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show rlir	Displays information about RLIR, Link Incident Record Registration (LIRR), and Distribute Registered Link Incident Record (DRLIR) frames.

debug rscn

To enable debugging for the registered state change notification (RSCN) feature, use the **debug rscn** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug rscn {**all** | **errors** | **events** | **mts-errors** | **mts-events**} [**vsan** *vsan-id*]

no debug rscn {**all** | **errors** | **events** | **mts-errors** | **mts-events**} [**vsan** *vsan-id*]

Syntax Description

all	Enables debugging for all RSCN features.
errors	Enables debugging for RSCN errors.
events	Enables debugging for RSCN events.
mts-errors	Enables debugging for RSCN MTS errors.
mts-events	Enables debugging for RSCN MTS events.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug rscn errors
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show rscn	Displays RSCN information.

debug san-ext-tuner

To enable debugging for SAN extension tuner, use the **debug san-ext-tuner** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug isns {**all** | **demux** | **deque** | **error** | **event** | **ha** | **trace** [**detail**] | **warning**}

no debug isns {**all** | **demux** | **deque** | **error** | **event** | **ha** | **trace** [**detail**] | **warning**}

Syntax Description

all	Enables all SAN extension tuner debugging.
demux	Enables debugging for SAN extension tuner message demux.
deque	Enables debugging for SAN extension tuner message dequeue.
error	Enables debugging for SAN extension tuner error conditions.
event	Enables debugging for SAN extension tuner events.
ha	Enables debugging for SAN extension tuner high availability.
trace	Enables debugging for SAN extension tuner trace.
detail	(Optional) Enables detailed debugging for SAN extension tuner trace.
warning	Enables debugging for SAN extension tuner warnings.

Defaults

None.

Command Modes

EXEC mode.

Command History

Release	Modification
2.0(x)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug san-ext-tuner error** command is issued:

```
switch# debug san-ext-tuner error
```

Related Commands

Command	Description
isns-server enable	Enables the iSNS server.
no debug all	Disables all debugging.

Command	Description
show isns	Displays iSNS information.
show san-ext-tuner	Displays SAN extension tuner information.

debug scsi-flow

To enable debugging of a SCSI flow, use the **debug scsi-flow** command. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug scsi-flow {**all** | **demux vsan** *vsan-id* | **deque** | **error** | **event vsan** *vsan-id* | **ha** | **trace** {**detail vsan** *vsan-id* | **vsan** *vsan-id*} | **warning vsan** *vsan-id*}

no debug scsi-flow {**all** | **demux vsan** *vsan-id* | **deque** | **error** | **event vsan** *vsan-id* | **ha** | **trace** {**detail vsan** *vsan-id* | **vsan** *vsan-id*} | **warning vsan** *vsan-id*}

Syntax Description

all	Enables all debug flags for all SCSI flows.
demux	Enables debugging for SCSI flow demux functions.
vsan *vsan-id*	Restricts debugging to the specified VSAN. The range is 1 to 4093.
deque	Enables debugging for SCSI flow deque events.
error	Enables debugging for SCSI flow errors.
event	Enables debugging for SCSI flow events.
ha	Enables debugging for SCSI flow high availability events.
trace	Enables debugging for SCSI flow traces.
detail	Enables debugging of SCSI flow detail trace.
warning	Enables debugging for SCSI flow warning messages.

Defaults

None.

Command Modes

EXEC mode.

Command History

Release	Modification
2.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

The following example enables all debug flags for all SCSI flows:

```
switch# debug scsi-flow all
2004 Nov 29 17:24:49 sfm: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
2004 Nov 29 17:24:49 sfm: fu_fsm_execute_all: null fsm_event_list
2004 Nov 29 17:24:49 sfm: fu_fsm_engine_post_event_processing: mts msg
MTS_OPC_DEBUG_WRAP_MSG(msg_id 536440) dropped
switch#
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show scsi-flow	Displays SCSI flow information.

debug scsi-target

To enable debugging for SCSI targets, use the **debug scsi-target** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug scsi-target {**error** | **flow**}

no debug scsi-target {**error** | **flow**}

Syntax Description

error	Enables debugging for SCSI target daemon error conditions.
flow	Enables debugging for the SCSI target flow.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.1(1)	This command was introduced.

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 vhbad: vhba_mts_handler: sdwrap_dispatch: retval:0
Apr 28 21:11:54 vhbad: vhbad_handle_timeout: timer:1 context:(nil)
Apr 28 21:12:06 vhbad: vhba_mts_handler: sysmgr_dispatch: retval:-1
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show scsi-target	Displays information about existing SCSI target configurations.

debug sdv

To enable debugging for SAN device virtualization, use the **debug sdv** command in EXEC mode.

debug sdv {**all** | **all-sdv** | **ddas** {**errors** | **events**} | **ddas-config** {**errors** | **events** | **packets**} | **discovery** {**errors vsan** *vsan-id* | **events vsan** *vsan-id*} | **distribution** {**errors vsan** *vsan-id* | **events vsan** *vsan-id*} | **errors vsan** *vsan-id* | **fu** {**ha** | **transition**} | **mgmt** {**errors** | **events**} | **ns** {**errors** | **events** | **packets**} | **rewrite** {**errors** | **events** | **packets**} | **trace vsan** *vsan-id* | **virtual-domain** {**errors vsan** *vsan-id* | **events vsan** *vsan-id*} | **zone-activation** {**errors** | **events** | **packets**}}

Syntax Description

all	Configures all SDV debugs.
all-sdv	Configures all filters for SDV debugging.
ddas	Enables the DDAS debugs.
errors	Enables debugs for errors.
events	Enables debugs for events.
ddas-config	Enables the DDAS-CFG debugs.
packets	Enables debugs for packets.
discovery	Enables the Disc debugs.
vsan *vsan-id*	Specifies the number of the VSAN. The range is 1 to 4093.
distribution	Enables the Dist debugs.
errors	Enables the Error debugs.
fu	Enables the FU debugs.
ha	Enables the FU HA debugs.
transition	Enables the transition debugs.
mgmt	Enables the Config FSM debugs.
ns	Enables the NS debugs.
rewrite	Enables the Rewrite debugs.
trace	Enables the Trace debugs.
virtual-domain	Enables the Virtual Domain debugs.
zone-activation	Enables the ZS-ACTV debugs.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
3.1(2)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug sdv all** command is issued.

```
switch# debug sdv all
2007 Jan 26 22:17:25.232055 sdv: fu_fsm_execute_all: match_msg_id(0), log_already_open(0)
2007 Jan 26 22:17:25.232151 sdv: fu_fsm_execute_all: null fsm_event_list
2007 Jan 26 22:17:25.232233 sdv: fu_fsm_engine_post_event_processing: mts msg
MTS_OPC_DEBUG_WRAP_MSG(msg_id 83409) dropped
```

Related Commands

Command	Description
sdv enable	Enables or disables SAN device virtualization.
show sdv statistics	Displays SAN device virtualization statistics.

debug security

To enable debugging for the security and accounting features, use the **debug security** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

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

no debug security {**all** | **events** | **mts** | **radius**}

Syntax Description

all	Enables debugging for all security features.
events	Enables debugging for security events.
mts	Enables debugging for security MTS packets.
radius	Enables debugging for RADIUS events.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug sensor

To enable debugging for the sensor manager, use the **debug sensor** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

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

no debug sensor {**demux** | **deque** | **error** | **info** | **init**}

Syntax Description

demux	Enables debugging for sensor demux functions.
deque	Enables debugging for sensor deque events.
error	Enables debugging for sensor errors.
info	Enables debugging for sensor information.
init	Enables debugging for sensor initialization.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
no debug all	Disables all debugging.
show environment temperature	Displays current temperature threshold settings and state.

debug sme

To enable debugging for the Cisco SME features, use the **debug sme** command. To disable a debug command, use the **no** form of the command.

debug sme {**all | demux vsan** *vsan id* **| deque | error | event vsan** *vsan id* **| ha vsan** *vsan id* **| trace vsan** *vsan id* **| trace-detail vsan** *vsan id* **| warning vsan** *vsan id*}

no debug sme {**all | demux vsan** *vsan id* **| deque | error | event vsan** *vsan id* **| ha vsan** *vsan id* **| trace vsan** *vsan id* **| trace-detail vsan** *vsan id* **| warning vsan** *vsan id*}

Syntax Description

all	Enables debugging of all Cisco SME features.
demux	Enables debugging of Cisco SME message demux.
vsan *vsan id*	Restricts debugging to a specified VSAN ID. The range is 1 to 4094.
deque	Enables debugging of Cisco SME message dequeue.
error	Enables debugging of Cisco SME errors.
event	Enables debugging of Cisco SME finite state machine (FSM) and events.
ha	Enables debugging of Cisco SME high availability (HA).
trace	Enables debugging of Cisco SME trace.
trace-detail	Enables debugging of Cisco SME trace-detail.
warning	Enables debugging of Cisco SME warning.

Defaults

None.

Command Modes

EXEC mode.

Command History

Release	Modification
3.2(2)	This command was introduced.

Usage Guidelines

None.

Examples

The following example displays the system output from the **debug sme all** command:

```
switch# debug sme all
2007 Sep 23 15:44:44.490796 sme: fu_priority_select:  - setting fd[5] for select
 call
2007 Sep 23 15:44:44.490886 sme: fu_priority_select_select_queue: round credit(8
)
2007 Sep 23 15:44:44.490918 sme:     curr_q - FU_PSEL_Q_CAT_CQ, usr_q_info(2), p
riority(7), credit(4), empty
2007 Sep 23 15:44:44.490952 sme: fu_priority_select: returning FU_PSEL_Q_CAT_MTS
 queue, fd(5), usr_q_info(1)
2007 Sep 23 15:44:44.491059 sme: sme_get_data_from_queue(1031): dequeued mts msg
```

```
 (34916564), MTS_OPC_DEBUG_WRAP_MSG
2007 Sep 23 15:44:44.491096 sme: fu_fsm_engine: line[2253]
2007 Sep 23 15:44:44.492596 sme: fu_fsm_execute_all: match_msg_id(0), log_alread
y_open(0)
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show sme	Displays all information about Cisco SME.

debug snmp

To enable debugging for the SNMP manager, use the **debug snmp** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

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

no debug snmp {**all** | **errors** | **mts** {**pkt** {**both** | **rx** [**node** *range* | **opcode** *range* | **sap** *range*] | **tx**} | **pkthdr** {**both** | **rx** [**numpkt** *range*] | **tx**}} | **pkt-dump** | **trace** {**trace-entryexit** | **trace-stub**}}

Syntax Description

all	Enables debugging for all SNMP output.
errors	Enables debugging for SNMP error output.
mts	Enables debugging for SNMP packets and headers.
pkt	Specifies debugging of packets.
both	Specifies debugging in both the transmit and receive directions.
rx	Specifies debugging in the receive direction.
node *range*	(Optional) Specifies the node for the packets in the receive direction. The integer range from 1 to 4095.
opcode *range*	(Optional) Specifies the opcode for the packets in the receive direction. The integer range from 1 to 4095.
sap *range*	(Optional)Specifies the SAP for the packets in the receive direction. The integer range from 1 to 4095.
tx	Specifies debugging in the transmit direction.
pkt	Specifies debugging of packets.
numpkt *range*	(Optional) Specifies the number of required packets.
trace	Enables trace level debug output.
trace-entryexit	Specifies trace-level entry or exit debug output.
trace-stub	Specifies trace-level stub debug output.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show snmp	Displays SNMP status and setting information.
snmp-server	Configures the SNMP server information, switch location, and switch name.
snmp-server enable traps	Enables SNMP server notifications (informs and traps).

debug span

To enable SPAN debugging, use the **debug span** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug span {**all** | **buffer-size** *bytes* | **error** | **event** | **trace** | **warning**}

no debug span {**all** | **error** | **event** | **trace** | **warning**}

Syntax Description

all	Enables debugging for all SPAN features.
buffer-size *bytes*	Configures event logs buffer size for SPAN. The range is 4096 to 131072.
error	Enables debugging for SPAN errors.
event	Enables debugging for SPAN events.
trace	Enables debugging for SPAN traces.
warning	Enables debugging for SPAN warning messages.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show span session	Displays specific information about a Switched Port Analyzer (SPAN) session.

debug system health

To enable system health monitoring debugging, use the **debug system health** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug system health {**all** | **asic-counters** | **battery-charger** | **bootflash** | **cache-disk** | **cfr** | **eobc** | **error** | **event** | **external-loopback** | **failure-analysis** | **fc2** | **free-disk** | **ha** | **inband** | **loopback** | **mgmt** | **misc** | **mts** | **nvram** | **plog** | **pss** | **serdes** | **special** | **trace** | **xipc**}

no debug system health {**all** | **asic-counters** | **battery-charger** | **bootflash** | **cache-disk** | **cfr** | **eobc** | **error** | **event** | **external-loopback** | **failure-analysis** | **fc2** | **free-disk** | **ha** | **inband** | **loopback** | **mgmt** | **misc** | **mts** | **nvram** | **plog** | **pss** | **serdes** | **special** | **trace** | **xipc**}

Syntax Description

all	Enables debugging of all online health flags.
asic-counters	Enables debugging of system health ASIC statistics.
battery-charger	Enables debugging of system health battery charger tests.
bootflash	Enables debugging of system health bootflash tests.
cache-disk	Enables debugging of system health cache-disk tests.
cfr	Enables debugging of system health compact health tests.
eobc	Enables debugging of system health EOBC tests.
error	Enables debugging of system health error conditions.
event	Enables debugging of system health events.
external-loopback	Enables debugging of system health external loopback tests.
failure-analysis	Enables debugging of system health failure analysis.
fc2	Enables debugging of system health FC2 frames.
free-disk	Enables debugging of system health free disk.
ha	Enables debugging of health monitoring HA flags.
inband	Enables debugging of system health inband tests.
loopback	Enables debugging of system health loopback tests.
mgmt	Enables debugging of system health management-port port tests.
misc	Enables debugging of system health misc.
mts	Enables debugging of system health MTS.
nvram	Enables debugging of system health nvram.
plog	Enables debugging of system health persistent logging.
pss	Enables debugging of system health pss.
serdes	Enables debugging of system health SerDes tests.
special	Enables debugging of system health special.
trace	Enables debugging of health monitoring trace flags.
xipc	Enables debugging of system health XIPC.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.
3.0(1)	Added the **free-disk**, **nvram**, and **plog** options.

Usage Guidelines

None.

Examples

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

```
switch# debug system health all
2005 Mar 10 01:49:28 SystemHealth: ohms_snake_fd_activity: Module 1 Snake Frame came.
2005 Mar 10 01:49:28 SystemHealth: ohms_snake_fd_activity: Module 8 waiting for Snake
Frame to come.
2005 Mar 10 01:49:28 SystemHealth: ohms_dequeue: select timeout 0 998000
2005 Mar 10 01:49:28 SystemHealth: fu_priority_select:  - setting fd[4] for select call  -
setting fd[20] for select call  - setting fd[22] for select call  - setting fd[28] for
select call  - setting fd[29] for select call  - setting fd[30] for select call
2005 Mar 10 01:49:28 SystemHealth: fu_priority_select_select_queue: round credit(14)
2005 Mar 10 01:49:28 SystemHealth:     curr_q - FU_PSEL_Q_CAT_FD, usr_q_info(466240),
fd(29), priority(6), credit(3), empty
2005 Mar 10 01:49:28 SystemHealth: fu_priority_select: returning FU_PSEL_Q_CAT_CQ queue,
usr_q_info(1)
2005 Mar 10 01:49:28 SystemHealth: ohms_dequeue: Select woken up
2005 Mar 10 01:49:28 SystemHealth: ohms_dequeue: Process event type 0x1
2005 Mar 10 01:49:28 SystemHealth: ohms_dequeue: Processing timer type
2005 Mar 10 01:49:28 SystemHealth: fu_fsm_engine: line[2139]
2005 Mar 10 01:49:28 SystemHealth: fu_fsm_handle_sysmgr_msg: Not mts event
2005 Mar 10 01:49:28 SystemHealth: ohms_timer_event_handler: called.
2005 Mar 10 01:49:28 SystemHealth: fu_fsm_execute_all: match_msg_id(0),
log_already_open(0)
.
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show system health	Displays configured Online Health Management System (OHMS) information.

debug tacacs+

To enable debugging for boot variables, use the **debug tacacs+** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug tacacs+ {**aaa-request** | **aaa-request-lowlevel** | **all** | **config** | **config-lowlevel** | **server-monitor** | **server-monitor-errors**}

no debug tacacs+ {**aaa-request** | **aaa-request-lowlevel** | **all** | **config** | **config-lowlevel** | **server-monitor** | **server-monitor-errors**}

Syntax Description

aaa-request	Enables TACACS+ AAA request debug.
aaa-request-lowlevel	Enables TACACS+ AAA request low-level debugging.
all	Enables all the debug flags.
config	Enables TACACS+ configuration debugging.
config-lowlevel	Enables TACACS+ configuring low-level debugging.
server-monitor	Enables TACACS+ server monitoring.
server-monitor-errors	Enables TACACS+ server monitor errors.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.3(1)	This command was introduced.
3.0(1)	Added the **server-monitor** and **server-monitor-errors** options.

Usage Guidelines

None.

Examples

The following example displays the system output when the **debug tacacs+ config-lowlevel** command is issued:

```
switch# debug tacacs+ config-lowlevel
Nov 20 06:39:44 tacacs: tacacs_debug_conf_open: entering...
172.22.94.252# Nov 20 06:39:44 tacacs: tacacs_debug_conf_open: exiting
Nov 20 06:39:44 tacacs: tacacs_conf_close: entering...
Nov 20 06:39:44 tacacs: tacacs_conf_close: returning 0
Nov 20 06:39:44 tacacs: tacacs_enable_info_config: entering for TACACS+ Daemon debug
Nov 20 06:39:44 tacacs: tacacs_debug_conf_open: entering...
Nov 20 06:39:44 tacacs: tacacs_debug_conf_open: exiting
Nov 20 06:39:44 tacacs: tacacs_enable_info_config: SET_REQ for TACACS+ Daemon debug with 1
Nov 20 06:39:44 tacacs: tacacs_enable_info_config: SET_REQ done for TACACS+ Daemon debug
with 1
```

```
Nov 20 06:39:44 tacacs: tacacs_enable_info_config: got back the return value of
configuration operation:success
Nov 20 06:39:44 tacacs: tacacs_debug_conf_close: entering...
Nov 20 06:39:44 tacacs: tacacs_debug_conf_close: returning 0
Nov 20 06:39:44 tacacs: tacacs_enable_info_config: exiting for TACACS+ Daemon debug
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show tacacs+	Displays the TACACS+ Cisco Fabric Services (CFS) distribution status and other details.

debug tcap

To enable debugging the exception logger, use the **debug tcap** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

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

no debug tcap {**demux** | **deque** | **error** | **info** | **init**}

Syntax Description

demux	Enables debugging for terminal capture demux functions.
deque	Enables debugging for terminal capture deque events.
error	Enables debugging for terminal capture errors.
info	Enables debugging for terminal capture information.
init	Enables debugging for terminal capture initialization.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

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

Examples

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

```
switch# debug tcap demux
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug tlport

To enable debugging for TL port interfaces, use the **debug tlport** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug tlport {**all** | **errors** | **events** {**fc2** {**terminal** | **transit**} | **mts** | **pss**}} [**interface fc** *slot/port*]

no debug tlport {**all** | **errors** | **events** {**fc2** {**terminal** | **transit**} | **mts** | **pss**}} [**interface fc** *slot/port*]

Syntax Description

all	Enables debugging for all TL port features.
errors	Enables debugging for TL port error conditions.
events	Enables debugging for TL port monitoring events.
fc2	Enables debugging for TL port monitoring FC 2 events.
terminal	Specifies TL port monitoring FC 2 terminating events.
transit	Specifies TL port monitoring FC 2 transit events.
mts	Enables debugging for TL port monitoring MTS packets.
pss	Enables debugging for TL port monitoring PSS packets.
interface fc *slot/port*	(Optional) Restricts debugging to the specified interface.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show tlport	Displays configured TL port information.

debug ttyd

To enable TTYD debugging, use the **debug ttyd** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug ttyd {**all** | **errors** | **events**}

no debug ttyd {**all** | **errors** | **events**}

Syntax Description

all	Enables debugging for all TTYD features.
errors	Enables debugging for TTYD error conditions.
events	Enables debugging for TTYD events.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug ttyd events
switch#
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug vni

To enable debugging for a virtual network interface (VNI), use the **debug vni** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

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

no debug vni {**all** | **errors** | **events** | **info** | **pss**}

Syntax Description

all	Enables debugging for all VNI features.
errors	Enables debugging for VNI error conditions.
events	Enables debugging for VNI events.
info	Enables debugging for VNI events.
pss	Enables debugging for VNI PSS packets.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug vrrp

To enable debugging for a Virtual Router Redundancy Protocol (VRRP), use the **debug vrrp** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug vrrp {**configuration** | **engine**} {**all** | **error** | **event** | **info**}

no debug vrrp {**configuration** | **engine**} {**all** | **error** | **event** | **info**}

Syntax Description

configuration	Enables VRRP configuration debugging.
engine	Enables VRRP engine debugging.
all	Enables debugging for all VRRP features.
error	Enables debugging for VRRP error conditions.
event	Enables debugging for VRRP events.
info	Enables debugging for VRRP events.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show vrrp	Displays VRRP configuration information.

debug vsan

To enable debugging for VSANs, use the **debug vsan** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug vsan {**all** | **global** | **ha** | **info** | **membership** | **mts**}

no debug vsan {**all** | **global** | **ha** | **info** | **membership** | **mts**}

Syntax Description

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

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug vsan all
2005 Mar 10 01:44:35 vsan: Calling handling function
2005 Mar 10 01:44:35 vsan: querying trunking membership(readonly) for interface:16859136
2005 Mar 10 01:44:35 vsan: Replying to trunking membership query for interface:fc1/21 with
VSAN bitmap:1-4093
2005 Mar 10 01:44:35 vsan: got back reply_code:0
2005 Mar 10 01:44:35 vsan: Returned from handling function
2005 Mar 10 01:44:35 vsan: Freeing notifications
2005 Mar 10 01:44:35 vsan: Src: 0x00000601/15  Dst: 0x00000601/27  ID: 0x0067CEA1  Size:
520 [RSP] Opc: 116 (MTS_OPC_VSAN_GET_PORT_TRUNKING_MEMBERSHIP) RR: 0x0067CEA0  HA_SEQNO:
0x00000000  TS: 0x24E717EAC7CE2  REJ:0  SYNC:1
2005 Mar 10 01:44:35 vsan: 00 00 00 00 00 00 02 00 7F FF FF FF FF FF FF FF
2005 Mar 10 01:44:35 vsan: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
2005 Mar 10 01:44:35 vsan: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
2005 Mar 10 01:44:35 vsan: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
2005 Mar 10 01:44:35 vsan: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
.
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show vsan	Displays information about configured VSANs.

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. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug wr-reg [*device-name* | *register-address*]

no debug wr-reg [*device-name* | *register-address*]

Syntax Description

device-name	(Optional) Specifies the device name for the required device.
register-address	(Optional) Specifies the register address for the required device.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

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

Related Commands

Command	Description
no debug all	Disables all debugging.

debug wwn

To enable debugging for the world wide name (WWN) manager, use the **debug wwn** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

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

no debug wwn {**all** | **detail** | **errors** | **flow** | **trace**}

Syntax Description

all	Enables all WWN debug options.
detail	Enables all WWN output
errors	Enables debugging for WWN error conditions.
flow	Enables flow-level WWN debug options.
trace	Enables debugging for WWN traces.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug wwn all
Apr 29 19:24:17 wwn: 53601-wwnm_sdwrap_dispatch:77|SDWRAP massage 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: 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 2E 00 00 00
```

```
Apr 29 19:24:17 wwn: 53601-wwnm_unmask_sigalrm:1261|TRACE:
FILE=_manager/wwnm/wwnm_utilities.c
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show wwn	Displays the status of the WWN configuration.

debug xbar

To enable crossbar debugging (XBAR), use the **debug xbar** command in EXEC mode. To disable a **debug** command, use the **no** form of the 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**}

no debug xbar {**all** | **demux** | **deque** | **error** [**module** *slot*] | **fsm** [**module** *slot*] | **ha** [**module** *slot*] | **init** | **main**}

Syntax Description

all	Enables all XBAR debug options.
demux	Enables debugging for XBAR demux functions.
deque	Enables debugging for XBAR deque events.
error	Enables debugging for XBAR errors.
module *slot*	(Optional) Specifies the slot number of the module being debugged.
fsm	Enables debugging for XBAR FSMs.
ha	Enables debugging for XBAR high availability information.
init	Enables debugging for XBAR initialization.
main	Enables XBAR debugging for main functions.

Defaults

Enabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
...
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug xbar_driver

To enable debugging of the crossbar driver (XBAR driver), use the **debug xbar_driver** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug xbar {**error** | **flow** | **trace**}

no debug xbar {**error** | **flow** | **trace**}

Syntax Description

error	Enables debugging of XBAR driver errors.
flow	Enables debugging of the XBAR driver flow.
trace	Enables debugging of the XBAR driver trace.

Defaults

Enabled.

Command Modes

EXEC mode.

Command History

Release	Modification
3.0(1)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug xbar_driver error
switch# 2006 Jan 23 22:02:41.770329 xbar_driver:  sc_stats_timer_hdlr  called
2006 Jan 23 22:03:41.780356 xbar_driver:  sc_stats_timer_hdlr  called
2006 Jan 23 22:04:41.780356 xbar_driver:  sc_stats_timer_hdlr  called
2006 Jan 23 22:05:41.780357 xbar_driver:  sc_stats_timer_hdlr  called
2006 Jan 23 22:06:41.780356 xbar_driver:  sc_stats_timer_hdlr  called
2006 Jan 23 22:07:41.780359 xbar_driver:  sc_stats_timer_hdlr  called
2006 Jan 23 22:08:41.790341 xbar_driver:  sc_stats_timer_hdlr  called...
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug xbc

To enable crossbar client debugging (XBC), use the **debug xbc** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug xbc {**demux** | **deque** | **init** | **main**}

no debug xbc {**demux** | **deque** | **init** | **main**}

Syntax Description

demux	Enables debugging for crossbar demux functions.
deque	Enables debugging for crossbar deque events.
init	Enables debugging for crossbar initialization.
main	Enables debugging for crossbar main functions.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

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
```

Related Commands

Command	Description
no debug all	Disables all debugging.

debug zone

To enable debugging for zones, use the **debug zone** command in EXEC mode. To disable a **debug** command, use the **no** form of the command or use the **no debug all** command to turn off all debugging.

debug zone {**all** | **change** {**errors** | **events** | **packets**} | **database** {**detail** | **errors** | **events**} **gs errors** {**errors** | **events** | **packets**} | **lun-zoning** {**errors** | **events** | **packets**} | **merge** {**errors** | **events** | **packets**} | **mts notifications** | **pss** {**errors** | **events**} | **read-only-zoning** {**errors** | **events** | **packets**} | **tcam errors** {**errors** | **events** | **packets**} | **transit** {**errors** | **events**}} [**vsan** *vsan-id*]

no debug zone {**all** | **change** {**errors** | **events** | **packets**} | **database** {**detail** | **errors** | **events**} | **gs errors** {**errors** | **events** | **packets**} | **lun-zoning** {**errors** | **events** | **packets**} | **merge** {**errors** | **events** | **packets**} | **mts notifications** | **pss** {**errors** | **events**} | **read-only-zoning** {**errors** | **events** | **packets**} | **tcam errors** {**errors** | **events** | **packets**} | **transit** {**errors** | **events**}} [**vsan** *vsan-id*]

Syntax Description

all	Enables all zone server debug options.
change	Enables debugging for change protocol messages.
errors	Enables debugging for zone errors.
events	Enables debugging for zone events.
packets	Enables debugging for zone packets.
database	
database	Enables debugging for database messages.
gs	Enables debugging for GS protocol messages.
lun-zoning	Enables debugging for LUN zoning messages.
merge	Enables debugging for merge protocol messages.
mts notification	Enables debugging for MTS notification messages.
pss	Enables debugging for PSS debug messages
read-only-zoning	Enables debugging for read-only Zoning messages.
tcam	Enables debugging for TCAM messages.
transit	Enables debugging for transit frame messages.
vsan *vsan-id*	(Optional) Restricts debugging to the specified VSAN.

Defaults

Disabled.

Command Modes

EXEC mode.

Command History

Release	Modification
1.0(2)	This command was introduced.

Usage Guidelines

None.

Examples

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

```
switch# debug zone all
2005 Mar 10 01:46:36 zone: Src: 0x00000601/18  Dst: 0x00000601/94  ID: 0x0067D5CD  Size:
276 [REQ] Opc: 182 (MTS_OPC_DEBUG_WRAP_MSG) RR: 0x0067D5CD  HA_SEQNO: 0x00000000  TS:
0x24E95060E0EF4  REJ:0  SYNC:0
2005 Mar 10 01:46:36 zone: 01 00 00 00 E8 03 00 00 00 00 00 00 00 00 00 00
2005 Mar 10 01:46:36 zone: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2005 Mar 10 01:46:36 zone: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2005 Mar 10 01:46:36 zone: FF FF FF FF 2F 64 65 76 2F 70 74 73 2F 30 00 00
2005 Mar 10 01:46:36 zone: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2005 Mar 10 01:46:36 zone: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.
```

Related Commands

Command	Description
no debug all	Disables all debugging.
show zone	Displays zone information.