debug netdr

To debug NetDriver activity, use the **debug netdr** command. Use the **no** form of this command to disable debugging output.

debug netdr {**all** | **data** | **error**}

no debug netdr {**all** | **data** | **error**}

Syntax Description

all	Debugs all NetDriver activity.
data	Debugs NetDriver data flow.
error	Debugs NetDriver errors.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(7)XE	Support for this command was introduced on the Cisco 7600 series routers.
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Examples

This example shows how to debug the NetDriver data flow:

```
Router# debug netdr data
NetDriver Receive Data on interrupt debugging is on
NetDriver Receive Data debugging is on
NetDriver Transmit Data debugging is on
NetDriver Relay Data debugging is on
Router#
2d21h: const_ether_vlan_vencap() Vlan1:
2d21h:    src_vlan=0x1  src_indx=0x3  len=0xE9  bpdu=0
2d21h:    index_dir=0  dest_indx=0x0  dont_lrn=0
2d21h:    Dbus hdr:  00000000 00010000 00030000 E9000000
2d21h:               00000000 00000000 00000000 00000000
2d21h:    MAC hdr:  dmac=00801C.938040, smac=00503E.8D6400, typelen=0800
2d21h:    IP hdr:  45C000DB 02F30000 FF066331 AC143412 AB45C8CC
2d21h: fx1000_process_receive_packet() Vlan1:
2d21h:    src_vlan=0x1  src_indx=0x108  len=0x40  bpdu=0
2d21h:    index_dir=0  dest_indx=0x3  dont_lrn=0
2d21h:    Dbus hdr:  60000000 00010000 01080000 40100000
2d21h:               0006AC14 3412AB45 C8CC0000 00030000
2d21h:    MAC hdr:  dmac=00503E.8D6400, smac=00605C.865B28, typelen=0800
2d21h:    IP hdr:  45000028 B5254000 7D06F471 AB45C8CC AC143412
<... output truncated ...>
Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs NetDriver continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture

To debug NetDriver capture activity, use the **debug netdr capture** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture [**and-filter** [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*]

no debug netdr capture [**and-filter** [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*]

Syntax Description

and-filter	(Optional) Applies filters.
destination-ip-address	(Optional) Captures all packets matching a destination IP address.
ipaddr	Captures packets for a specific destination IP address.
ipv6 *ipaddr*	Captures all packets matching the IPv6 destination IP address.
dmac *mac-addr*	(Optional) Captures packets matching a destination MAC address index.
dstindex *index-value*	(Optional) Captures all packets matching a destination index; valid values are 0 to 1048575.
ethertype *ethertype*	(Optional) Captures all packets matching an ethertype; ethertype must be entered in hexidecimal format.
interface *interface*	(Optional) Captures packets related to the interface. See Usage Guidelines.
smac *smac*	(Optional) Captures packets matching the source MAC address; smac must be entered in hexidecimal format.
source-ip-address	(Optional) Captures all packets matching a source IP address.
srcindex *index-value*	(Optional) Captures all packets matching a source index; valid values are 0 to 1048575.
vlan *vlan-num*	(Optional) Captures packets matching the VLAN number; valid VLAN numbers are 0 to 4095.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Usage Guidelines

You can use the following interface types:

- Async
- Auto-template
- CTunnel
- Dialer
- EsconPhy
- Fcpa
- Filter
- Filtergroup
- GMPLS
- GigabitEthernet
- Group-Async
- LISP
- LongReachEthernet
- Looopback
- Lspvif
- MFR
- Multilink
- Null
- Port-channel
- Sysclock
- TenGigabitEthernet
- Tunnel
- Vif
- Virtual-Ethernet
- Virtual-Template
- Virtual-TokenRing
- VLAN
- VoaBypassIn
- VoaBypassOut
- VoaFilterIn
- VoaFilterOut
- VoaIn
- VoaOut

Examples

This example shows how to debug the NetDriver:

```
Router# debug netdr capture

Router#
```

Related Commands

Command	Description
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture and-filter

To debug NetDriver capture activity using an **and** function, use the **debug netdr capture and-filter** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture **and-filter** [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*]

no debug netdr capture and-filter [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*]

Syntax Description

destination-ip-address	(Optional) Captures all packets matching a destination IP address.
ipaddr	Captures packets for a specific destination IP address.
ipv6 *ipaddr*	Captures all packets matching the IPv6 destination IP address.
dmac *mac-addr*	(Optional) Captures packets matching a destination MAC address index.
dstindex *index-value*	(Optional) Captures all packets matching a destination index; valid values are 0 to 1048575.
ethertype *ethertype*	(Optional) Captures all packets matching an ethertype; ethertype must be entered in hexidecimal format.
interface *interface*	(Optional) Captures packets related to the interface. See Usage Guidelines.
smac *smac*	(Optional) Captures packets matching the source MAC address; smac must be entered in hexidecimal format.
source-ip-addr ess	(Optional) Captures all packets matching a source IP address.
srcindex *index-value*	(Optional) Captures all packets matching a source index; valid values are 0 to 1048575.
vlan *vlan-num*	(Optional) Captures packets matching the VLAN number; valid VLAN numbers are 0 to 4095.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Usage Guidelines

You can use the following interface types:

- Async
- Auto-template
- CTunnel
- Dialer
- EsconPhy
- Fcpa
- Filter
- Filtergroup
- GMPLS
- GigabitEthernet
- Group-Async
- LISP
- LongReachEthernet
- Looopback
- Lspvif
- MFR
- Multilink
- Null
- Port-channel
- Sysclock
- TenGigabitEthernet
- Tunnel
- Vif
- Virtual-Ethernet
- Virtual-Template
- Virtual-TokenRing
- VLAN
- VoaBypassIn
- VoaBypassOut
- VoaFilterIn
- VoaFilterOut
- VoaIn
- VoaOut

Examples

This example shows how to debug the NetDriver:

```
Router# debug netdr capture

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture continuous

To debug NetDriver capture activity continuously, use the **debug netdr capture continuous** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture continuous [**and-filter** | **destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **or-filter** [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*] | **rx** [**and-filter** | **destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dti-type** *value* | **dti-value** *value* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **or-filter** [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*] | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*]

no debug netdr capture continuous [**and-filter** | **destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **or-filter** [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*] | **rx** [**and-filter** | **destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dti-type** *value* | **dti-value** *value* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **or-filter** [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*] | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*]

Syntax Description

and-filter	(Optional) Applies filters.
destination-ip-address	(Optional) Captures all packets matching a destination IP address.
ipaddr	Captures packets for a specific destination IP address.
ipv6 *ipaddr*	Captures all packets matching the IPv6 destination IP address.
dmac *mac-addr*	(Optional) Captures packets matching a destination MAC address index.
dstindex *index-value*	(Optional) Captures all packets matching a destination index; valid values are 0 to 1048575.
ethertype *ethertype*	(Optional) Captures all packets matching an ethertype; ethertype must be entered in hexidecimal format.
interface *interface*	(Optional) Captures packets related to the interface. See Usage Guidelines.
or-filter	(Optional) Applies filters.
rx	(Optional) Captures incoming packets only.
dti-type *value*	(Optional) Captures all packets matching the 3-bit dti type; valid values are 0 to 7.
dti-value *value*	(Optional) Captures all packets matching the 21-bit dti value; valid values are 0 to 4096.
smac *smac*	(Optional) Captures packets matching the source MAC address; smac must be entered in hexidecimal format.

source-ip-address	(Optional) Captures all packets matching a source IP address.
srcindex *index-value*	(Optional) Captures all packets matching a source index; valid values are 0 to 1048575.
vlan *vlan-num*	(Optional) Captures packets matching the VLAN number; valid VLAN numbers are 0 to 4095.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Usage Guidelines

You can use the following interface types:

- Async
- Auto-template
- CTunnel
- Dialer
- EsconPhy
- Fcpa
- Filter
- Filtergroup
- GMPLS
- GigabitEthernet
- Group-Async
- LISP
- LongReachEthernet
- Looopback
- Lspvif
- MFR
- Multilink
- Null
- Port-channel
- Sysclock
- TenGigabitEthernet

- Tunnel
- Vif
- Virtual-Ethernet
- Virtual-Template
- Virtual-TokenRing
- VLAN
- VoaBypassIn
- VoaBypassOut
- VoaFilterIn
- VoaFilterOut
- VoaIn
- VoaOut

Examples

This example shows how to debug the NetDriver:

```
Router# debug netdr capture


Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture destination-ip-address

To debug NetDriver capture activity capturing all packets matching a destination IP address, use the **debug netdr capture destination-ip-address** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture destination-ip-address {*ipaddr* | **ipv6** *ipaddr*}

no debug netdr capture destination-ip-address {*ipaddr* | **ipv6** *ipaddr*}

Syntax Description

ipaddr	Captures packets for a specific destination IP address.
ipv6 *ipaddr*	Captures all packets matching the IPv6 destination IP address.

Defaults

None

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Command Modes

Privileged EXEC

Examples

This example shows how to debug the NetDriver:

```
Router# debug netdr capture

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture dmac

To debug NetDriver capture activity by capturing all matching destination MAC addresses, use the **debug netdr capture dmac** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture dmac [*mac-addr*]

no debug netdr capture dmac [*mac-addr*]

Syntax Description

mac-addr	(Optional) Captures packets matching a destination MAC address index.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Examples

This example shows how to debug the NetDriver:

```
Router# debug netdr capture


Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.

Command	Description
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture dstindex

To debug NetDriver capture activity capturing all packets matching the destination index, use the **debug netdr capture dstindex** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture dstindex [*index-value*]

no debug netdr capture dstindex [*index-value*]

Syntax Description

index-value	(Optional) Captures all packets matching a destination index; valid values are 0 to 1048575.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Examples

This example shows how to debug the NetDriver:

```
Router# debug netdr capture

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.

Command	Description
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture ethertype

To debug NetDriver capture ethertype activity, use the **debug netdr capture ethertype** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output. .

debug netdr capture ethertype [*ethertype*]

no debug netdr capture ethertype [*ethertype*]

Syntax Description

ethertype	(Optional) Captures all packets matching an ethertype; ethertype must be entered in hexidecimal format.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Examples

This example shows how to debug the NetDriver ethertype:

```
Router# debug netdr capture ethertype

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.

Command	Description
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture interface

To debug NetDriver capture interface activity, use the **debug netdr capture interface** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture interface [*interface*]

no debug netdr capture interface [*interface*]

Syntax Description

interface	(Optional) Captures packets related to the interface. See Usage Guidelines.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Usage Guidelines

You can use the following interface types:

- Async
- Auto-template
- CTunnel
- Dialer
- EsconPhy
- Fcpa
- Filter
- Filtergroup
- GMPLS
- GigabitEthernet
- Group-Async
- LISP
- LongReachEthernet
- Looopback
- Lspvif
- MFR

- Multilink
- Null
- Port-channel
- Sysclock
- TenGigabitEthernet
- Tunnel
- Vif
- Virtual-Ethernet
- Virtual-Template
- Virtual-TokenRing
- VLAN
- VoaBypassIn
- VoaBypassOut
- VoaFilterIn
- VoaFilterOut
- VoaIn
- VoaOut

Examples

This example shows how to debug the NetDriver interface activity:

```
Router# debug netdr capture interface

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.

Command	Description
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture or-filter

To debug NetDriver capture activity using an **or** function, use the **debug netdr capture or-filter** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture or-filter [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*]

no debug netdr capture or-filter [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*]

Syntax Description

destination-ip-address	(Optional) Captures all packets matching a destination IP address.
ipaddr	Captures packets for a specific destination IP address.
ipv6 *ipaddr*	Captures all packets matching the IPv6 destination IP address.
dmac *mac-addr*	(Optional) Captures packets matching a destination MAC address index.
dstindex *index-value*	(Optional) Captures all packets matching a destination index; valid values are 0 to 1048575.
ethertype *ethertype*	(Optional) Captures all packets matching an ethertype; ethertype must be entered in hexidecimal format.
interface *interface*	(Optional) Captures packets related to the interface. See Usage Guidelines.
smac *smac*	(Optional) Captures packets matching the source MAC address; smac must be entered in hexidecimal format.
source-ip-addr ess	(Optional) Captures all packets matching a source IP address.
srcindex *index-value*	(Optional) Captures all packets matching a source index; valid values are 0 to 1048575.
vlan *vlan-num*	(Optional) Captures packets matching the VLAN number; valid VLAN numbers are 0 to 4095.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Usage Guidelines

You can use the following interface types:

- Async
- Auto-template
- CTunnel
- Dialer
- EsconPhy
- Fcpa
- Filter
- Filtergroup
- GMPLS
- GigabitEthernet
- Group-Async
- LISP
- LongReachEthernet
- Looopback
- Lspvif
- MFR
- Multilink
- Null
- Port-channel
- Sysclock
- TenGigabitEthernet
- Tunnel
- Vif
- Virtual-Ethernet
- Virtual-Template
- Virtual-TokenRing
- VLAN
- VoaBypassIn
- VoaBypassOut
- VoaFilterIn
- VoaFilterOut
- VoaIn
- VoaOut

Examples

This example shows how to debug the NetDriver or-filter:

```
Router# debug netdr capture or-filter

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture rx

To debug NetDriver capture activity by capturing incoming packets only, use the **debug netdr capture rx** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture rx [**dti-type** *value* | **dti-value** *value*]

no debug netdr capture rx [**dti-type** *value* | **dti-value** *value*]

Syntax Description

dti-type *value*	(Optional) Captures all packets matching the 3-bit dti type; valid values are 0 to 7.
dti-value *value*	(Optional) Captures all packets matching the 21-bit dti value; valid values are 0 to 4096.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Examples

This example shows how to debug the NetDrivers incoming packets:

```
Router# debug netdr capture rx


Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.

Command	Description
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture smac

To debug NetDriver capture activity by capturing matching source MAC addresses, use the **debug netdr capture smac** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture smac [*smac*]

no debug netdr capture smac [*smac*]

Syntax Description

smac	(Optional) Captures packets matching the source MAC address; smac must be entered in hexidecimal format.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Examples

This example shows how to debug the NetDriver by capturing the source MAC addresses:

```
Router# debug netdr capture smac

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.

Command	Description
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture source-ip-address

To debug NetDriver capture activity by capturing all packets matching a source IP address, use the **debug netdr capture source-ip-address** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture source-ip-address {*ipaddr* | **ipv6** *ipaddr*}

no debug netdr capture source-ip-address {*ipaddr* | **ipv6** *ipaddr*}

Syntax Description

ipaddr	Captures packets for a specific destination IP address.
ipv6 *ipaddr*	Captures all packets matching the IPv6 destination IP address.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Examples

This example shows how to debug the NetDriver: source IP address

```
Router# debug netdr capture source-ip-address

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.

Command	Description
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture srcindex

To debug NetDriver capture activity by capturing all packets matching the source index, use the **debug netdr capture srcindex** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture srcindex [*index-value*]

no debug netdr capture srcindex [i*ndex-value*]

Syntax Description

index-value	(Optional) Captures all packets matching a source index; valid values are 0 to 1048575.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Examples

This example shows how to debug the NetDriver by capturing all packets matching the source index:

```
Router# debug netdr capture srcindex

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.

Command	Description
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture tx

To debug NetDriver capture activity by capturing the outgoing packets only, use the **debug netdr capture tx** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture tx [**and-filter** | **destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **or-filter** [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*]

no debug netdr capture tx [**andand-filter** | **destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **dmac** *mac-addr* | **dstindex** *index-value* | **ethertype** *ethertype* | **interface** *interface* | **or-filter** [**destination-ip-address** {*ipaddr* | **ipv6** *ipaddr*}| **smac** *smac* | **source-ip-address** {*ipaddr* | **ipv6** *ipaddr*} | **srcindex** *index-value* | **vlan** *vlan-num*]

Syntax Description

and-filter	(Optional) Captures all added filters.
destination-ip-address	(Optional) Captures all packets matching a destination IP address.
ipaddr	Captures packets for a specific destination IP address.
ipv6 *ipaddr*	Captures all packets matching the IPv6 destination IP address.
dmac *mac-addr*	(Optional) Captures packets matching a destination MAC address index.
dstindex *index-value*	(Optional) Captures all packets matching a destination index; valid values are 0 to 1048575.
ethertype *ethertype*	(Optional) Captures all packets matching an ethertype; ethertype must be entered in hexidecimal format.
interface *interface*	(Optional) Captures packets related to the interface. See Usage Guidelines.
or-filter	(Optional) Applies filters.
smac *smac*	(Optional) Captures packets matching the source MAC address; smac must be entered in hexidecimal format.
source-ip-addr ess	(Optional) Captures all packets matching a source IP address.
srcindex *index-value*	(Optional) Captures all packets matching a source index; valid values are 0 to 1048575.
vlan *vlan-num*	(Optional) Captures packets matching the VLAN number; valid VLAN numbers are 0 to 4095.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Usage Guidelines

You can use the following interface types:

- Async
- Auto-template
- CTunnel
- Dialer
- EsconPhy
- Fcpa
- Filter
- Filtergroup
- GMPLS
- GigabitEthernet
- Group-Async
- LISP
- LongReachEthernet
- Looopback
- Lspvif
- MFR
- Multilink
- Null
- Port-channel
- Sysclock
- TenGigabitEthernet
- Tunnel
- Vif
- Virtual-Ethernet
- Virtual-Template
- Virtual-TokenRing
- VLAN
- VoaBypassIn
- VoaBypassOut
- VoaFilterIn
- VoaFilterOut
- VoaIn

- VoaOut

Examples

This example shows how to debug the NetDriver:

```
Router# debug netdr capture tx

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr capture vlan

To debug NetDriver capture activity by capturing packets matching a specific VLAN number, use the **debug netdr capture vlan** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr capture vlan [*vlan-num*]

no debug netdr capture vlan [*vlan-num*]

Syntax Description

vlan-num	(Optional) Captures packets matching the VLAN number; valid VLAN numbers are 0 to 4095.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Examples

This example shows how to debug the NetDriver by capturing packets matching a specific VLAN number:

```
Router# debug netdr capture vlan 10

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.

Command	Description
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr clear-capture	Clears the capture buffer.
debug netdr copy-captured	Copies the packets to a file.

debug netdr clear-capture

To clear the capture buffer, use the **debug netdr clear-capture** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr clear-capture

no debug netdr clear-capture

Syntax Description

This command has no keywords or arguments.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Examples

This example shows how to debug the NetDriver:

```
Router# debug netdr clear-capture

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.

Command	Description
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr copy-captured	Copies the packets to a file.

debug netdr copy-captured

To store captured packets to a file, use the **debug netdr copy-captured** command in Privileged EXEC mode. Use the **no** form of this command to disable debugging output.

debug netdr copy-captured

no debug netdr copy-captured

Syntax Description

This command has no keywords or arguments.

Defaults

None

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(14)SX	Support for this command on the Cisco 7600 series routers was extended to the 12.1 E release.

Usage Guidelines

You can copy a captured file to the following sources:

- bootdisk:
- const_nvram:
- dfc#2-bootflash:
- dfc#3-bootflash:
- disk0:
- ftp:
- http:
- https:
- image:
- null:
- nvram:
- rcp:
- scp:
- syslog:
- tftp:
- tmpsys:

Examples

This example shows how to debug the NetDriver copied packets:

```
Router# debug netdr copy-captured

Router#
```

Related Commands

Command	Description
debug netdr capture	Debugs NetDriver capture activity.
debug netdr capture and-filter	Debugs added filters.
debug netdr capture continuous	Debugs netdr continuously.
debug netdr capture destination-ip-address	Debugs all matching destination packets.
debug netdr capture dmac	Debugs matching destination packets.
debug netdr capture dstindex	Debugs packets matching destination index.
debug netdr capture ethertype	Debugs packets matching the ethertype.
debug netdr capture interface	Debugs packets related to an interface.
debug netdr capture or-filter	Debugs or-filter function packets.
debug netdr capture rx	Debugs incoming packets only.
debug netdr capture smac	Debugs packets matching the source MAC address.
debug netdr capture source-ip-address	Debugs packets matching the source IP address.
debug netdr capture srcindex	Debugs packets matching the source index.
debug netdr capture tx	Debugs outgoing packets only.
debug netdr capture vlan	Debugs packets for a specific VLAN.
debug netdr clear-capture	Clears the capture buffer.

lldp run

To enable processing of received LLDP control packets and enable transmission of LLDP packets with default or configured TLVs, use the **lldp run** command in Global configuration mode.

lldp run

Syntax Description

This command has no keywords or arguments.

Defaults

LLDP is disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SXJ	Support for this command was introduced.

Usage Guidelines

Configuring this command enables LLDP protocol on the switch. Unconfiguring it disables processing or transmit of LLDP protocol packets from the switch.

Examples

This example shows how to enable LLDP on the switch:

```
Switch(config)# lldp run
```

Related Commands

Command	Description
lldp tlv-select power-management	Enables power negotiation through LLDP.

lldp tlv-select power-management

To to enable power negotiation through LLDP, use the **lldp tlv-select power-management** Global configuration command.

lldp tlv-select power-management

Syntax Description

This command has no keywords or arguments.

Defaults

Enabled on POEP ports

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SXJ	Support for this command was introduced.

Usage Guidelines

You need to disable this feature if you do not want to perform power negotiation through LLDP.

This feature is not supported on non-POEP ports; the CLI is suppressed on such ports and TLV is not exchanged.

Examples

This example shows how to enable LLDP power negotiation on interface Gigabit Ethernet 3/1:

```
Switch# configuration t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 3/1
Switch(config-if)# lldp tlv-select power-management
```

Related Commands

Command	Description
lldp run	Enables processing of received LLDP control packets and enable transmission of LLDP packets with default or configured TLVs.

mls acl tcam share-acl

To enable sharing of ACLs, use the **mls acl tcam share-acl** command in Global configuration mode. To turn off sharing of the global defaults, use the **no** form of this command.

mls acl tcam share-acl

no mls acl tcam share-acl

Syntax Description

This command has no arguments or keywords.

Defaults

Enabled

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SXH	Support for this command was introduced.

Usage Guidelines

If you power cycle one of the DFCs, we recommend that you reset all the DFCs across the ACLs of the different DFCs.

Examples

This example shows how to enable sharing of ACLs:

```
Router(config)# mls acl tcam share-acl
```

mvr (global configuration)

To enable the multicast VLAN registration (MVR) feature on the switch, use the **mvr** global configuration command without keywords on the switch stack or on a standalone switch. Use the **no** form of this command to return to the default settings.

mvr [**group** *ip-address* [*count*] | **mode** [**compatible** | **dynamic**] | **querytime** *value* | **vlan** *vlan-id*]

no mvr [**group** *ip-address* | **mode** [**compatible** | **dynamic**] | **querytime** *value* | **vlan** *vlan-id*]

Syntax Description

group *ip-address*	(Optional) Statically configures an MVR group IP multicast address on the switch. Use the **no** form of this command to remove a statically configured IP multicast address or contiguous addresses or, when no IP address is entered, to remove all statically configured MVR IP multicast addresses.
count	(Optional) Configures multiple contiguous MVR group addresses. The range is 1 to 256; the default is 1.
mode	(Optional) Specifies the MVR mode of operation. The default is compatible mode.
compatible	(Optional) Sets MVR mode to provide compatibility with Catalyst 2900 XL and Catalyst 3500 XL switches. This mode does not allow dynamic membership joins on source ports.
dynamic	(Optional) Sets MVR mode to allow dynamic MVR membership on source ports.
querytime *value*	(Optional) Sets the maximum time to wait for IGMP report memberships on a receiver port. This time applies only to receiver-port leave processing. When an IGMP query is sent from a receiver port, the switch waits for the default or configured MVR querytime for an IGMP group membership report before removing the port from multicast group membership. The value is the response time in units of tenths of a second. The range is 1 to 100; the default is 5 tenths or one-half second. Use the **no** form of the command to return to the default setting.
vlan *vlan-id*	(Optional) Specifies the VLAN on which MVR multicast data is expected to be received. This is also the VLAN to which all the source ports belong. The range is 1 to 4094; the default is VLAN 1.

Defaults

MVR is disabled by default.

The default MVR mode is compatible mode.

No IP multicast addresses are configured on the switch by default.

The default group IP address count is 0.

The default query response time is 5 tenths of or one-half second.

The default multicast VLAN for MVR is VLAN 1.

Command Modes Global configuration

Command History

Release	Modification
12.2(33)SXH	This command was introduced.

Usage Guidelines Use the **mvr** command with keywords to set the MVR mode for a switch, configure the MVR IP multicast address, set the maximum time to wait for a query reply before removing a port from group membership, and to specify the MVR multicast VLAN. A maximum of 256 MVR multicast groups can be configured on a switch.

Use the **mvr group** command to statically set up all the IP multicast addresses that will take part in MVR. Any multicast data sent to a configured multicast address is sent to all the source ports on the switch and to all receiver ports that have registered to receive data on that IP multicast address.

MVR supports aliased IP multicast addresses on the switch. However, if the switch is interoperating with Catalyst 6500 Series switches, you should not configure IP addresses that create an alias between themselves or with the reserved IP multicast addresses (in the range 224.0.0.xxx).

The **mvr querytime** command applies only to receiver ports.

If the switch MVR is interoperating with Catalyst 6500 Series switches, set the multicast mode to compatible.

When operating in compatible mode, MVR does not support IGMP dynamic joins on MVR source ports.

MVR can coexist with IGMP snooping on a switch.

Multicast routing and MVR cannot coexist on a switch. If you enable multicast routing and a multicast routing protocol while MVR is enabled, MVR is disabled and a warning message appears. If you try to enable MVR while multicast routing and a multicast routing protocol are enabled, the operation to enable MVR is cancelled and an Error message is displayed.

Examples This example shows how to enable MVR:

```
Switch(config)# mvr
```

Use the **show mvr** privileged EXEC command to display the current setting for maximum multicast groups.

This example shows how to configure 228.1.23.4 as an IP multicast address:

```
Switch(config)# mvr group 228.1.23.4
```

This example shows how to configure ten contiguous IP multicast groups with multicast addresses from 228.1.23.1 to 228.1.23.10:

```
Switch(config)# mvr group 228.1.23.1 10
```

Use the **show mvr members** privileged EXEC command to display the IP multicast group addresses configured on the switch.

This example shows how to set the maximum query response time as one second (10 tenths):

```
Switch(config)# mvr querytime 10
```

This example shows how to set VLAN 2 as the multicast VLAN:

```
Switch(config)# mvr vlan 2
```

You can verify your settings by entering the **show mvr** privileged EXEC command.

Related Commands

Command	Description
mvr (interface configuration)	Configures MVR ports.
show mvr	Displays MVR global parameters or port parameters.
show mvr interface	Displays the configured MVR interfaces with their type, status, and Immediate Leave configuration. Also displays all MVR groups of which the interface is a member.
show mvr members	Displays all ports that are members of an MVR multicast group; if the group has no members, its status is shown as Inactive.

mvr (interface configuration)

To configure a Layer 2 port as a multicast VLAN registration (MVR) receiver or source port, to set the Immediate Leave feature, and to statically assign a port to an IP multicast VLAN and IP address, use the **mvr** interface configuration command on the switch stack or on a standalone switch. Use the **no** form of this command to return to the default settings.

mvr [**immediate** | **type** {**receiver** | **source**} | **vlan** *vlan-id* **group** [*ip-address*]]

no mvr [**immediate** | **type** {**source** | **receiver**}| **vlan** *vlan-id* **group** [*ip-address*]]

Syntax Description

immediate	(Optional) Enables the Immediate Leave feature of MVR on a port. Use the **no mvr immediate** command to disable the feature.
type	(Optional) Configures the port as an MVR receiver port or a source port. The default port type is neither an MVR source nor a receiver port. The **no mvr type** command resets the port as neither a source or a receiver port.
receiver	Configures the port as a subscriber port that can only receive multicast data. Receiver ports cannot belong to the multicast VLAN.
source	Configures the port as an uplink port that can send and receive multicast data for the configured multicast groups. All source ports on a switch belong to a single multicast VLAN.
vlan *vlan-id* **group**	(Optional) Adds the port as a static member of the multicast group with the specified VLAN ID. The **no mvr vlan** *vlan-id* **group** command removes a port on a VLAN from membership in an IP multicast address group.
ip-address	(Optional) Statically configures the specified MVR IP multicast group address for the specified multicast VLAN ID. This is the IP address of the multicast group that the port is joining.

Defaults

A port is configured as neither a receiver nor a source.

The Immediate Leave feature is disabled on all ports.

No receiver port is a member of any configured multicast group.

Command Modes

Interface configuration

Command History

Release	Modification
12.2(33)SXH	This command was introduced.

Usage Guidelines

Configure a port as a source port if that port should be able to both send and receive multicast data bound for the configured multicast groups. Multicast data is received on all ports configured as source ports.

Receiver ports cannot be trunk ports. Receiver ports on a switch can be in different VLANs, but should not belong to the multicast VLAN.

A port that is not taking part in MVR should not be configured as an MVR receiver port or a source port. A non-MVR port is a normal switch port, able to send and receive multicast data with normal switch behavior.

When Immediate Leave is enabled, a receiver port leaves a multicast group more quickly. Without Immediate Leave, when the switch receives an IGMP leave message from a group on a receiver port, it sends out an IGMP MAC-based query on that port and waits for IGMP group membership reports. If no reports are received in a configured time period, the receiver port is removed from multicast group membership. With Immediate Leave, an IGMP MAC-based query is not sent from the receiver port on which the IGMP leave was received. As soon as the leave message is received, the receiver port is removed from multicast group membership, which speeds up leave latency.

The Immediate Leave feature should be enabled only on receiver ports to which a single receiver device is connected.

The **mvr vlan group** command statically configures ports to receive multicast traffic sent to the IP multicast address. A port statically configured as a member of group remains a member of the group until statically removed. In compatible mode, this command applies only to receiver ports; in dynamic mode, it can also apply to source ports. Receiver ports can also dynamically join multicast groups by using IGMP join messages.

When operating in compatible mode, MVR does not support IGMP dynamic joins on MVR source ports.

An MVR port cannot be a private-VLAN port.

Examples

This example shows how to configure a port as an MVR receiver port:

```
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# mvr type receiver
```

Use the **show mvr interface** privileged EXEC command to display configured receiver ports and source ports.

This example shows how to enable Immediate Leave on a port:

```
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# mvr immediate
```

This example shows how to add a port on VLAN 1 as a static member of IP multicast group 228.1.23.4:

```
Switch(config)# interface gigabitethernet1/0/2
Switch(config-if)# mvr vlan1 group 230.1.23.4
```

You can verify your settings by entering the **show mvr members** privileged EXEC command.

Related Commands

Command	Description
mvr (global configuration)	Enables and configures multicast VLAN registration on the switch.
show mvr	Displays MVR global parameters or port parameters.
show mvr interface	Displays the configured MVR interfaces or displays the multicast groups to which a receiver port belongs. Also displays all MVR groups of which the interface is a member.
show mvr members	Displays all receiver ports that are members of an MVR multicast group.

platform system-controller reset-threshold

To configure the system controller reset threshold, use the **platform system-controller reset-threshold** command.

platform system-controller reset threshold {*threshold-num*}

Syntax Description

threshold-num	Specifies the threshold reset number; valid values are 1 to 100.

Defaults

System controller reset is set to 1.

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SXJ4	Support for this command was introduced.
12.2(33)SXI10	Support for this command was introduced.

Usage Guidelines

If you have a redundant supervisor engine and a TM_DATA_PARITY_ERROR, TM_LINK_ERR_INBAND, or TM_NPP_PARITY_ERROR error occurs, the affected supervisor engine reloads. When you do not have a redundant supervisor engine and a TM_DATA_PARITY_ERROR, TM_LINK_ERR_INBAND, or TM_NPP_PARITY_ERROR error occurs, one of the following happens:

- If the system controller reset threshold has not been reached, the system controller ASIC resets.
- If the system controller reset threshold has been reached, the supervisor engine reloads.

If the threshold has not been reached you will see these system messages:

```
%SYSTEM_CONTROLLER-<>-THRESHOLD
%SYSTEM_CONTROLLER-<>-ERROR
%SYSTEM_CONTROLLER-<>-MISTRAL_RESET
```

If the threshold has been reached you will see these system messages:

```
%SYSTEM_CONTROLLER-<>-ERROR
%SYSTEM_CONTROLLER-<>-FATAL
```

Examples

This example shows how to configure the system controller reset threshold to 55:

```
Router(config)# platform system-controller reset-threshold 55
```

show mvr

To display the current Multicast VLAN Registration (MVR) global parameter values, including whether or not MVR is enabled, the MVR multicast VLAN, the maximum query response time, the number of multicast groups, and the MVR mode (dynamic or compatible), use the **show mvr** privileged EXEC command.

show mvr

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SXH	This command was introduced.

Examples

This is an example of output from the **show mvr** command:

```
Switch# show mvr
MVR Running: TRUE
MVR multicast VLAN: 1
MVR Max Multicast Groups: 256
MVR Current multicast groups: 0
MVR Global query response time: 5 (tenths of sec)
MVR Mode: compatible
```

In the preceding display, the maximum number of multicast groups is fixed at 256. The MVR mode is either compatible (for interoperability with Catalyst 2900 XL and Catalyst 3500 XL switches) or dynamic (where operation is consistent with IGMP snooping operation and dynamic MVR membership on source ports is supported).

Related Commands

Command	Description
mvr (global configuration)	Enables and configures multicast VLAN registration on the switch.
mvr (interface configuration)	Configures MVR ports.
show mvr interface	Displays the configured MVR interfaces, status of the specified interface, or all multicast groups to which the interface belongs when the **interface** and **members** keywords are appended to the command.
show mvr members	Displays all ports that are members of an MVR multicast group or, if there are no members, means the group is inactive.

show mvr interface

To display the Multicast VLAN Registration (MVR) receiver and source ports, use the **show mvr interface** privileged EXEC command without keywords . Use the command with keywords to display MVR parameters for a specific receiver port.

show mvr interface [*interface-id* [**members** [**vlan** *vlan-id*]]]

Syntax Description

interface-id	(Optional) Displays MVR type, status, and Immediate Leave setting for the interface; valid interfaces include physical ports (including type, stack member [stacking-capable switches only] module, and port number).
members	(Optional) Displays all MVR groups to which the specified interface belongs.
vlan *vlan-id*	(Optional) Displays all MVR group members on this VLAN. The range is 1 to 4094.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SXH	This command was introduced.

Usage Guidelines

If the entered port identification is a non-MVR port or a source port, the command returns an error message. For receiver ports, it displays the port type, per port status, and Immediate-Leave setting.

If you enter the **members** keyword, all MVR group members on the interface appear. If you enter a VLAN ID, all MVR group members in the VLAN appear.

Examples

This is an example of output from the **show mvr interface** command:

```
Switch# show mvr interface
Port        Type            Status          Immediate Leave
----        ----            -------         ---------------
Gi1/0/1      SOURCE          ACTIVE/UP       DISABLED
Gi1/0/2      RECEIVER        ACTIVE/DOWN     DISABLED
```

In the preceding display, Status is defined as follows:

- Active means the port is part of a VLAN.
- Inactive means that the port is not yet part of any VLAN.
- Up/Down means that the port is forwarding/nonforwarding.

This is an example of output from the **show mvr interface** command for a specified port:

```
Switch# show mvr interface gigabitethernet1/0/2
Type: RECEIVER Status: ACTIVE Immediate Leave: DISABLED
```

This is an example of output from the **show mvr interface** *interface-id* **members** command:

```
Switch# show mvr interface gigabitethernet1/0/2 members
239.255.0.0     DYNAMIC ACTIVE
239.255.0.1     DYNAMIC ACTIVE
239.255.0.2     DYNAMIC ACTIVE
239.255.0.3     DYNAMIC ACTIVE
239.255.0.4     DYNAMIC ACTIVE
239.255.0.5     DYNAMIC ACTIVE
239.255.0.6     DYNAMIC ACTIVE
239.255.0.7     DYNAMIC ACTIVE
239.255.0.8     DYNAMIC ACTIVE
239.255.0.9     DYNAMIC ACTIVE
```

Related Commands

Command	Description
mvr (global configuration)	Enables and configures multicast VLAN registration on the switch.
mvr (interface configuration)	Configures MVR ports.
show mvr	Displays the global MVR configuration on the switch.
show mvr members	Displays all receiver ports that are members of an MVR multicast group.

show mvr members

To display all receiver and source ports that are currently members of an IP multicast group, use the **show mvr members** privileged EXEC command.

show mvr members [*ip-address*]

Syntax Description

ip-address	(Optional) The IP multicast address. If the address is entered, all receiver and source ports that are members of the multicast group appear. If no address is entered, all members of all Multicast VLAN Registration (MVR) groups are listed. If a group has no members, the group is listed as Inactive.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SXH	This command was introduced.

Usage Guidelines

The **show mvr members** command applies to receiver and source ports. For MVR-compatible mode, all source ports are members of all multicast groups.

Examples

This example shows the status of all mvr members:

```
Switch# show mvr members
MVR Group IP    Status          Members
------------    ------          -------
239.255.0.1     ACTIVE          Gi1/0/1(d), Gi1/0/5(s)
239.255.0.2     INACTIVE        None
239.255.0.3     INACTIVE        None
239.255.0.4     INACTIVE        None
239.255.0.5     INACTIVE        None
239.255.0.6     INACTIVE        None
239.255.0.7     INACTIVE        None
239.255.0.8     INACTIVE        None
239.255.0.9     INACTIVE        None
239.255.0.10    INACTIVE        None

<output truncated>
```

This example shows the status of an IP address and the members of the IP multicast group with that IP address:

```
Switch# show mvr members 239.255.0.2
239.255.003.--22     ACTIVE          Gi1//1(d), Gi1/0/2(d), Gi1/0/3(d),
                                     Gi1/0/4(d), Gi1/0/5(s)
```

Related Commands

Command	Description
mvr (global configuration)	Enables and configures multicast VLAN registration on the switch.
mvr (interface configuration)	Configures MVR ports.
show mvr	Displays the global MVR configuration on the switch.
show mvr interface	Displays the configured MVR interfaces, status of the specified interface, or all multicast groups to which the interface belongs when the **members** keyword is appended to the command.

show platform hardware cef tcam

To display platform hardware Cisco Express Forwarding (CEF) Forwarding Information Base (FIB) Ternary Content Addressable Memory (TCAM), use the **show platform hardware cef** command.

show platform hardware cef tcam {**ecc** [**detail** [**earl** *earl-id*] | **module** *module-num*]| **earl** *earl-id*]| **module** *module-num*]| **hit** [**detail** [**earl** *earl-id*] | **module** *module-num*]| **earl** *earl-id*]| **module** *module-num*]| **keys** [**count** | **exception**]| **memory usage** | **segment** [**detail** [**earl** *earl-id*] | **module** *module-num*]| **earl** *earl-id*]| **module** *module-num*] | **select** [**detail** [**earl** *earl-id*] | **module** *module-num*]| **earl** *earl-id*]| **module** *module-num*] | **shadow** [**detail** [**earl** *earl-id*] | **module** *module-num*]| **earl** *earl-id*]| **module** *module-num*] | **timing** [**detail** [**earl** *earl-id*] | **module** *module-num*]| **earl** *earl-id*]| **module** *module-num*] | **utilization** [**detail** [**earl** *earl-id*] | **module** *module-num*]| **earl** *earl-id*]| **module** *module-num*]| **earl** *earl-id*] | **module** *module-num*]

Syntax Description

ecc	Displays error checking and correction (ECC) information.
detail	(Optional) Displays detailed information.
earl *earl-id*	(Optional) Displays earl-id content.
module *module-num*	(Optional) Displays information for a specific module.
hit	Displays last hit on the FIB TCAM information.
keys	Displays keys information.
count	(Optional) Displays keys count information.
exception	(Optional) Displays keys exception iformation.
memory usage	Displays memory usage.
segment	Displays segment distribution.
select	Displays bit-select information.
shadow	Displays the shadow copy.
timing	Displays timing ultilization.
utilization	Displays segment ultilization.

Defaults

None

Command Modes

Privileged EXEC mode

Command History

Release	Modification
12.2(14)SX	Support for this command was introduced.

Examples

This example shows how to display the hardware CEF TCAM key exception:

```
Router(config)# show platform hardware cef tcam keys exception
Priorities in exception:
 Class          ID   Pri (>=)   Max   Key-Cnt   Pri-Cnt
```

```
  IPv4           0    16          35    35085
  IPv4-Mcast     1    68          68    0
  MPLS           2    17          17    0
  EOMPLS         3    19          19    0
  MPLS-VPN       4    9           9     0
  Diags          5    5           5     0
  IPv6-Local     6    390         390   0
  IPv6-Mcast     7    261         261   0
  IPv6-Global    8    244         390   1051
  VPLSv4-Mcast   9    69          69    0
  VPLSv6-Mcast   10   261         261   0

Keys in each Pri in exception:
  Class          ID   Pri   XCP   Pri-Cnt
  IPv4           0    16          4096
    .            .    17          15507
    .            .    18          7753
    .            .    19          3876
    .            .    20          1939
    .            .    21          969
    .            .    22          484
    .            .    23          243
    .            .    24          121
    .            .    25          60
    .            .    26          34
    .            .    30          2
    .            .    34          1
  IPv6-Global    8    244         129
    .            .    245         126
    .            .    246         118
    .            .    247         114
    .            .    248         111
    .            .    249         109
    .            .    250         109
    .            .    251         107
    .            .    252         64
    .            .    253         32
    .            .    254         16
```

```
.            .    255         8
.            .    256         4
.            .    257         2
.            .    389         1
.            .    390         1
```

```
Spanslogic#show platform hardware cef tcam memory usage
Buffer allocation summary:
      Data  Page    Total Total   Used  Used    Free  Free
Id    Size  Size    Count Size    Count Size    Count Size    Type
0     0     0       0     0       0     0       0     0       Void
1     80    80      0     0       0     0       0     0       Bsort
2     2076  65536   279   576K    0     0       279   565K    Bsort Node
3     3456  65536   288   1.00M   20    39744   268   904K    Bsort Stat
4     60    65536   2184  128K    1697  99.4K   487   29220   Wsort Seg
5     104   65536   630   65536   17    1768    613   63752   Wsort Win
6     1024  65536   384   384K    17    17408   367   367K    Wsort Avail
7     3644  3644    3     10932   3     10932   0     0       Group
8     324   8192    25    8192    20    6480    5     1620    Group Entry
9     0     0       2     20480   2     20480   0     0       SE Block
10    4104  65536   3660  15.2M   2814  11.1M   846   3.31M   SE Slice
11    52    65536   2520  128K    1697  88244   823   42796   SE Seg
12    68    65536   18297 1.18M   0     0       18297 1.18M   SE Rec
13    0     0       0     0       0     0       0     0       SE Pri
14    48    65536   619k  28.3M   587k  26.8M   32320 1.47M   Key
15    8     65536   0     0       0     0       0     0       Bit 64
16    12    65536   1.20m 13.7M   1.13m 13.2M   63352 742K    Bit 96
17    16    65536   0     0       0     0       0     0       Bit 128
18    20    65536   39312 768K    36634 715K    2678  53560   Bit 160
19    24    24      0     0       0     0       0     0       Bit 192
20    28    28      0     0       0     0       0     0       Bit 224
21    32    65536   8192  256K    5628  175K    2564  82048   Bit 256
22    36    65536   0     0       0     0       0     0       Bit 288
23    40    65536   1638  65536   98    3920    1540  61600   Bit 320
24    44    44      0     0       0     0       0     0       Bit 352
25    48    48      0     0       0     0       0     0       Bit 384
26    52    52      0     0       0     0       0     0       Bit 416
```

```
27   56    56       0     0       0     0       0     0       Bit 448
28   60    65536    0     0       0     0       0     0       Bit 480
29   64    64       0     0       0     0       0     0       Bit 512
30   68    68       0     0       0     0       0     0       Bit 544
31   72    72       0     0       0     0       0     0       Bit 576
32   76    76       0     0       0     0       0     0       Bit 608
33   80    65536    0     0       0     0       0     0       Bit 640
34   84    84       0     0       0     0       0     0       Bit 672
35   88    88       0     0       0     0       0     0       Bit 704
*    *     *        1.89m 61.9M   1.77m 52.7M   124k  8.81M   Total
Router(config)#
```

This example shows how to display the hardware CEF TCAM timing information:

```
Router(config)# show platform hardware cef tcam timing
(0) Groom Clock: 182us (Min) << 71ms (Avg) << 404ms (Max) = 09.19s (Tot)
  5.28ms    1.16ms    10ms      38ms      3.43ms    10ms      166ms     223ms
  5.06ms    960us     9.34ms    37ms      1.79ms    96ms      110ms     155ms
  4.72ms    1.06ms    8.90ms    34ms      813us     4.14ms    110ms     138ms
  4.12ms    755us     6.81ms    32ms      305us     75ms      28ms      237ms
  3.90ms    690us     6.13ms    30ms      228us     50ms      247ms     199ms
  3.67ms    529us     5.81ms    28ms      274us     94ms      217ms     63ms
  3.38ms    506us     3.68ms    25ms      1.73ms    269ms     218ms     96ms
  3.14ms    400us     2.94ms    23ms      279us     119ms     277ms     66ms
  2.94ms    351us     3.63ms    20ms      115ms     163ms     346ms     94ms
  2.87ms    306us     2.75ms    18ms      46ms      404ms     316ms     35ms
  2.76ms    291us     1.18ms    16ms      156ms     351ms     154ms     84ms
  3.30ms    275us     794us     13ms      87ms      319ms     220ms     5.51ms
  5.20ms    202us     736us     11ms      8.52ms    85ms      220ms     203ms
  2.77ms    190us     39ms      9.58ms    112ms     229ms     189ms     191ms
  1.58ms    182us     39ms      7.52ms    73ms      180ms     172ms     216ms
  1.84ms    11ms      38ms      5.63ms    90ms      188ms     227ms     1.27ms

(1) Add Clock: 11us (Min) << 13us (Avg) << 107us (Max) = 1.78ms (Tot)
  11us      11us      11us      12us      11us      12us      12us      12us
  11us      12us      12us      11us      11us      11us      11us      12us
  12us      11us      12us      12us      11us      12us      12us      12us
  12us      12us      12us      12us      11us      12us      12us      13us
  11us      12us      12us      12us      11us      11us      11us      12us
  12us      12us      12us      12us      11us      12us      12us      12us
  12us      12us      11us      12us      11us      11us      11us      12us
  11us      11us      12us      11us      11us      12us      12us      12us
  11us      12us      11us      11us      11us      12us      11us      11us
  12us      12us      12us      11us      11us      11us      11us      40us
```

```
  107us      12us      12us      12us      12us      12us      12us      41us
  12us       11us      11us      12us      12us      12us      11us      40us
  12us       12us      11us      12us      12us      11us      11us      40us
  11us       11us      12us      12us      12us      12us      11us      40us
  12us       12us      12us      12us      12us      11us      12us      40us
  12us       12us      11us      11us      11us      12us      12us      40us
Router(config)#
```

This example shows how to display the hardware CEF TCAM utilization information:

```
Router(config)# show platform hardware cef tcam utilization
Util summary for Pool 0: 524288 keys, 1024 segs, 36 Mb

Type  KeyCnt   KeyUse   SegCnt  SegUse  Util  Free
0     463704   463704   909     909     99    115
1     0        0        0       0       0     115
2     0        0        0       0       0     57
3     0        0        0       0       0     29
4     0        0        0       0       0     28
Tot   463704   463704   909     909     99    115

Util summary for Pool 1: 524288 keys, 1024 segs, 36 Mb

Type  KeyCnt   KeyUse   SegCnt  SegUse  Util  Free
0     105327   105327   208     208     98    803
1     9        18       7       7       0     803
2     46       184      3       6       5     391
3     0        0        0       0       0     191
4     0        0        0       0       0     189
Tot   105382   105529   218     221     93    803

Util summary for Pool 8: 1048576 keys, 2048 segs, 72 Mb

Type  KeyCnt   KeyUse   SegCnt  SegUse  Util  Free
0     569031   569031   1117    1117    99    918
1     9        18       7       7       0     918
2     46       184      3       6       5     448
3     0        0        0       0       0     220
4     0        0        0       0       0     217
Tot   569086   569233   1127    1130    98    918
```

Related Commands

Command	Description
clear platform hardware cef adjacencies	Clears platform hardware CEF adjacencies.

snmp-server enable traps fru-control

To enable CISCO-ENTITY-FRU-CONTROL-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the **snmp-server enable traps fru-control** command in privileged EXEC mode. To disable CISCO-ENTITY-SENSOR-MIB, use the **no** form of this command.

snmp-server enable traps fru-control

no snmp-server enable traps fru-control

Syntax Description

This command has no arguments or keywords.

Command Default

This command has no defaults.

Command Modes

Privileged EXEC mode

Command History

Release	Modification
12.2(1)SX	This command was introduced.

Examples

This example shows how to enable the CISCO-ENTITY-FRU-CONTROL-MIB MIB:

```
Switch# snmp-server enable traps fru-control
Switch#
```

This example shows how to disable the CISCO-ENTITY-FRU-CONTROL-MIB MIB:

```
Switch# no snmp-server enable traps fru-control
Switch#
```

Related Commands

Command	Description
snmp-server enable traps module	Enables CISCO-STACK-MIB Simple Network Management Protocol (SNMP) notifications.

snmp-server enable traps module

To enable CISCO-STACK-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the **snmp-server enable traps module** command in privileged EXEC mode. To disable CISCO-STACK-MIB, use the **no** form of this command.

snmp-server enable traps module

no snmp-server enable traps module

Syntax Description

This command has no arguments or keywords.

Command Default

This command has no defaults.

Command Modes

Privileged EXEC mode

Command History

Release	Modification
12.2(1)SX	This command was introduced.

Examples

This example shows how to enable the CISCO-STACK-MIB MIB:

```
Router# snmp-server enable traps module
Router#
```

This example shows how to disable the CISCO-STACK-MIB MIB:

```
Router# no snmp-server enable traps module
Router#
```

Command	Description
snmp-server enable traps fru-control	Enables CISCO-ENTITY-FRU-CONTROL-MIB Simple Network Management Protocol (SNMP) notifications.