



## Catalyst 3750 Switch Debug Commands

---

This appendix describes the **debug** privileged EXEC commands that have been created or changed for use with the Catalyst 3750 switch. These commands are helpful in diagnosing and resolving internetworking problems and should be enabled only under the guidance of Cisco technical support staff.



### Caution

---

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use the **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. It is best to use the **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use.

---

# debug auto qos

Use the **debug auto qos** privileged EXEC command to enable debugging of the automatic quality of service (auto-QoS) feature. Use the **no** form of this command to disable debugging output.

**debug auto qos**

**no debug auto qos**

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

**Defaults** Auto-QoS debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(14)EA1	This command was introduced.
	12.2(18)SE	The <b>debug auto qos</b> command replaced the <b>debug autoqos</b> command.

**Usage Guidelines** To display the QoS configuration that is automatically generated when auto-QoS is enabled, enable debugging *before* you enable auto-QoS. You enable debugging by entering the **debug auto qos** privileged EXEC command.

The **undebug auto qos** command is the same as the **no debug auto qos** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

**Examples** This example shows how to display the QoS configuration that is automatically generated when auto-QoS is enabled:

```
Switch# debug auto qos
AutoQoS debugging is on
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet2/0/1
Switch(config-if)# auto qos voip cisco-phone

21:29:41: mls qos map cos-dscp 0 8 16 26 32 46 48 56
21:29:41: mls qos
21:29:42: no mls qos srr-queue input cos-map
21:29:42: no mls qos srr-queue output cos-map
21:29:42: mls qos srr-queue input cos-map queue 1 threshold 3 0
21:29:42: mls qos srr-queue input cos-map queue 1 threshold 2 1
```

```

21:29:42: mls qos srr-queue input cos-map queue 2 threshold 1 2
21:29:42: mls qos srr-queue input cos-map queue 2 threshold 2 4 6 7
21:29:43: mls qos srr-queue input cos-map queue 2 threshold 3 3 5
21:29:43: mls qos srr-queue output cos-map queue 1 threshold 3 5
21:29:43: mls qos srr-queue output cos-map queue 2 threshold 3 3 6 7
21:29:44: mls qos srr-queue output cos-map queue 3 threshold 3 2 4
21:29:44: mls qos srr-queue output cos-map queue 4 threshold 2 1
21:29:44: mls qos srr-queue output cos-map queue 4 threshold 3 0
21:29:44: no mls qos srr-queue input dscp-map
21:29:44: no mls qos srr-queue output dscp-map
21:29:44: mls qos srr-queue input dscp-map queue 1 threshold 2 9 10 11 12 13 14 15
21:29:45: mls qos srr-queue input dscp-map queue 1 threshold 3 0 1 2 3 4 5 6 7
21:29:45: mls qos srr-queue input dscp-map queue 1 threshold 3 32
21:29:45: mls qos srr-queue input dscp-map queue 2 threshold 1 16 17 18 19 20 21 22 23
21:29:45: mls qos srr-queue input dscp-map queue 2 threshold 2 33 34 35 36 37 38 39 48
21:29:46: mls qos srr-queue input dscp-map queue 2 threshold 2 49 50 51 52 53 54 55 56
21:29:46: mls qos srr-queue input dscp-map queue 2 threshold 2 57 58 59 60 61 62 63
21:29:46: mls qos srr-queue input dscp-map queue 2 threshold 3 24 25 26 27 28 29 30 31
21:29:47: mls qos srr-queue input dscp-map queue 2 threshold 3 40 41 42 43 44 45 46 47
21:29:47: mls qos srr-queue output dscp-map queue 1 threshold 3 40 41 42 43 44 45 46 47
21:29:47: mls qos srr-queue output dscp-map queue 2 threshold 3 24 25 26 27 28 29 30 31
21:29:47: mls qos srr-queue output dscp-map queue 2 threshold 3 48 49 50 51 52 53 54 55
21:29:48: mls qos srr-queue output dscp-map queue 2 threshold 3 56 57 58 59 60 61 62 63
21:29:48: mls qos srr-queue output dscp-map queue 3 threshold 3 16 17 18 19 20 21 22 23
21:29:48: mls qos srr-queue output dscp-map queue 3 threshold 3 32 33 34 35 36 37 38 39
21:29:49: mls qos srr-queue output dscp-map queue 4 threshold 1 8
21:29:49: mls qos srr-queue output dscp-map queue 4 threshold 2 9 10 11 12 13 14 15
21:29:49: mls qos srr-queue output dscp-map queue 4 threshold 3 0 1 2 3 4 5 6 7
21:29:49: no mls qos srr-queue input priority-queue 1
21:29:49: no mls qos srr-queue input priority-queue 2
21:29:50: mls qos srr-queue input bandwidth 90 10
21:29:50: no mls qos srr-queue input buffers
21:29:50: mls qos queue-set output 1 buffers 10 10 26 54
21:29:50: interface GigabitEthernet2/0/1
21:29:50: mls qos trust device cisco-phone
21:29:50: mls qos trust cos
21:29:50: no queue-set 1
21:29:50: srr-queue bandwidth shape 10 0 0 0
21:29:50: srr-queue bandwidth share 10 10 60 20

```

**Related Commands**

Command	Description
<a href="#">auto qos voip</a>	Configures auto-QoS for voice over IP (VoIP) within a QoS domain.
<a href="#">show auto qos</a>	Displays the initial configuration that is generated by the automatic auto-QoS feature
<a href="#">show debugging</a>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug cluster

Use the **debug cluster** privileged EXEC command to enable debugging of cluster-specific events. Use the **no** form of this command to disable debugging output.

```
debug cluster { discovery | events | extended | hsrp | http | ip [packet] | members | nat | neighbors
| platform | snmp | vqpxy }
```

```
no debug cluster { discovery | events | extended | hsrp | http | ip [packet] | members | nat |
neighbors | platform | snmp | vqpxy }
```

## Syntax Description

<b>discovery</b>	Display cluster discovery debug messages.
<b>events</b>	Display cluster event debug messages.
<b>extended</b>	Display extended discovery debug messages.
<b>hsrp</b>	Display the Hot Standby Router Protocol (HSRP) debug messages.
<b>http</b>	Display Hypertext Transfer Protocol (HTTP) debug messages.
<b>ip</b> [ <b>packet</b> ]	Display IP or transport packet debug messages.
<b>members</b>	Display cluster member debug messages.
<b>nat</b>	Display Network Address Translation (NAT) debug messages.
<b>neighbors</b>	Display cluster neighbor debug messages.
<b>platform</b>	Display platform-specific cluster debug messages.
<b>snmp</b>	Display Simple Network Management Protocol (SNMP) debug messages.
<b>vqpxy</b>	Display VLAN Query Protocol (VQP) proxy debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

This command is available only on the cluster command switch stack or cluster command switch.

The **undebug cluster** command is the same as the **no debug cluster** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session** *switch-number* privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command** *stack-member-number* *LINE* privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<b>show cluster</b>	Displays the cluster status and a summary of the cluster to which the switch belongs.
	<b>show cluster candidates</b>	Displays a list of candidate switches when entered on the command switch.
	<b>show cluster members</b>	Displays information about cluster members when executed on the command switch.

# debug dot1x

Use the **debug dot1x** privileged EXEC command to enable debugging of the 802.1x feature. Use the **no** form of this command to disable debugging output.

**debug dot1x** {all | errors | events | packets | registry | state-machine}

**no debug dot1x** {all | errors | events | packets | registry | state-machine}

## Syntax Description

<b>all</b>	Display all 802.1x debug messages.
<b>errors</b>	Display 802.1x error debug messages.
<b>events</b>	Display 802.1x event debug messages.
<b>packets</b>	Display 802.1x packet debug messages.
<b>registry</b>	Display 802.1x registry invocation debug messages.
<b>state-machine</b>	Display state-machine related-events debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.
12.1(14)EA1	The <b>authsm</b> , <b>backend</b> , <b>besm</b> , <b>core</b> , and <b>reauthsm</b> keywords were removed. The <b>errors</b> , <b>events</b> , <b>packets registry</b> , and <b>state-machine</b> keywords were added.

## Usage Guidelines

The **undebug dot1x** command is the same as the **no debug dot1x** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the
	<a href="#">show dot1x</a>	Displays 802.1x statistics, administrative status, and operational status for the switch or for the specified port.

# debug dtp

Use the **debug dtp** privileged EXEC command to enable debugging of the Dynamic Trunking Protocol (DTP) activity. Use the **no** form of this command to disable debugging output.

**debug dtp** { **aggregation** | **all** | **decision** | **events** | **oserrs** | **packets** | **queue** | **states** | **timers** }

**no debug dtp** { **aggregation** | **all** | **decision** | **events** | **oserrs** | **packets** | **queue** | **states** | **timers** }

## Syntax Description

<b>aggregation</b>	Display DTP user-message aggregation debug messages.
<b>all</b>	Display all DTP debug messages.
<b>decision</b>	Display the DTP decision-table debug messages.
<b>events</b>	Display the DTP event debug messages.
<b>oserrs</b>	Display DTP operating system-related error debug messages.
<b>packets</b>	Display DTP packet-processing debug messages.
<b>queue</b>	Display DTP packet-queueing debug messages.
<b>states</b>	Display DTP state-transition debug messages.
<b>timers</b>	Display DTP timer-event debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug dtp** command is the same as the **no debug dtp** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.



Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<b>show dtp</b>	Displays DTP information for the switch or for a specified interface.

# debug etherchannel

Use the **debug etherchannel** privileged EXEC command to enable debugging of the EtherChannel/PAGP shim. This shim is the software module that is the interface between the Port Aggregation Protocol (PAgP) software module and the port manager software module. Use the **no** form of this command to disable debugging output.

**debug etherchannel** [**all** | **detail** | **error** | **event** | **idb**]

**no debug etherchannel** [**all** | **detail** | **error** | **event** | **idb**]

## Syntax Description

<b>all</b>	(Optional) Display all EtherChannel debug messages.
<b>detail</b>	(Optional) Display detailed EtherChannel debug messages.
<b>error</b>	(Optional) Display EtherChannel error debug messages.
<b>event</b>	(Optional) Debug major EtherChannel event messages.
<b>idb</b>	(Optional) Display PAgP interface descriptor block debug messages.



## Note

Though visible in the command-line help strings, the **linecard** keyword is not supported.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

If you do not specify a keyword, all debug messages appear.

The **undebug etherchannel** command is the same as the **no debug etherchannel** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<b>show etherchannel</b>	Displays EtherChannel information for the channel.

# debug ilpower

Use the **debug ilpower** privileged EXEC command to enable debugging of the power controller and Power over Ethernet (PoE) system.

```
debug ilpower { controller | event | port | registries }
```

```
no debug ilpower { controller | event | port | registries }
```



## Note

This command is supported only on PoE-capable switches.

## Syntax Description

<b>controller</b>	Display PoE controller debug messages.
<b>event</b>	Display PoE event debug messages.
<b>port</b>	Display PoE port manager debug messages.
<b>registries</b>	Display PoE registries debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(19)EA1	This command was introduced.

## Usage Guidelines

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<a href="#">show controllers power inline</a>	Displays the values in the registers of the specified PoE controller.
<a href="#">show power inline</a>	Displays the power status for the specified PoE port or for all PoE ports.

# debug ip dhcp snooping

Use the **debug ip dhcp snooping** privileged EXEC command to enable debugging of DHCP snooping. Use the **no** form of this command to disable debugging output.

**debug ip dhcp snooping** {*mac-address* | **agent** | **event** | **packet**}

**no debug ip dhcp snooping** {*mac-address* | **agent** | **event** | **packet**}

This command is available only if your switch is running the enhanced multilayer image (EMI).

Syntax Description		
<i>mac-address</i>	Display debug messages for a DHCP packet with the specified MAC address.	
<b>agent</b>	Display debug messages for DHCP snooping agents.	
<b>event</b>	Display debug messages for DHCP snooping events.	
<b>packet</b>	Display debug messages for DHCP snooping.	

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(20)SE	This command was introduced.

**Usage Guidelines** The **undebug ip dhcp snooping** command is the same as the **no debug ip dhcp snooping** command. When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select <b>Cisco IOS Release 12.2 Configuration Guides and Command References &gt; Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2 &gt; System Management &gt; Troubleshooting and Fault Management</b> .

# debug ip verify source packet

Use the **debug ip verify source packet** privileged EXEC command to enable debugging of IP source guard. Use the **no** form of this command to disable debugging output.

**debug ip verify source packet**

**no debug ip verify source packet**

This command is available only if your switch is running the enhanced multilayer image (EMI).

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

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.2(20)SE	This command was introduced.

**Usage Guidelines** The **undebug ip verify source packet** command is the same as the **no debug ip verify source packet** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select <b>Cisco IOS Release 12.2 Configuration Guides and Command References &gt; Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2 &gt; System Management &gt; Troubleshooting and Fault Management</b> .

# debug interface

Use the **debug interface** privileged EXEC command to enable debugging of interface-related activities. Use the **no** form of this command to disable debugging output.

```
debug interface { interface-id | null interface-number | port-channel port-channel-number |
vlan vlan-id }
```

```
no debug interface { interface-id | null interface-number | port-channel port-channel-number |
vlan vlan-id }
```

Syntax Description		
<i>interface-id</i>		Display debug messages for the specified physical port, identified by type switch number/module number/ port, for example <b>gigabitethernet 1/0/2</b> .
<b>null</b> <i>interface-number</i>		Display debug messages for null interfaces. The <i>interface-number</i> is always <b>0</b> .
<b>port-channel</b> <i>port-channel-number</i>		Display debug messages for the specified EtherChannel port-channel interface. The <i>port-channel-number</i> range is 1 to 12.
<b>vlan</b> <i>vlan-id</i>		Display debug messages for the specified VLAN. The <i>vlan-id</i> range is 1 to 4094.

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** If you do not specify a keyword, all debug messages appear.

The **undebug interface** command is the same as the **no debug interface** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session** *switch-number* privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command** *stack-member-number* *LINE* privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## ■ debug interface

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<b>show etherchannel</b>	Displays EtherChannel information for the channel.



# debug ip igmp filter

Use the **debug ip igmp filter** privileged EXEC command to enable debugging of Internet Group Management Protocol (IGMP) filter events. Use the **no** form of this command to disable debugging output.

**debug ip igmp filter**

**no debug ip igmp filter**

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

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebug ip igmp filter** command is the same as the **no debug ip igmp filter** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug ip igmp max-groups

Use the **debug ip igmp max-groups** privileged EXEC command to enable debugging of Internet Group Management Protocol (IGMP) maximum groups events. Use the **no** form of this command to disable debugging output.

**debug ip igmp max-groups**

**no debug ip igmp max-groups**

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

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebug ip igmp max-groups** command is the same as the **no debug ip igmp max-groups** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug ip igmp snooping

Use the **debug igmp snooping** privileged EXEC command to enable debugging of Internet Group Management Protocol (IGMP) snooping activity. Use the **no** form of this command to disable debugging output.

```
debug ip igmp snooping [group | management | router | timer]
```

```
no debug ip igmp snooping [group | management | router | timer]
```

## Syntax Description

<b>group</b>	(Optional) Display IGMP snooping group activity debug messages.
<b>management</b>	(Optional) Display IGMP snooping management activity debug messages.
<b>router</b>	(Optional) Display IGMP snooping router activity debug messages.
<b>timer</b>	(Optional) Display IGMP snooping timer event debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug ip igmp snooping** command is the same as the **no debug ip igmp snooping** command. When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<a href="#">debug platform ip igmp snooping</a>	Displays information about platform-dependent IGMP snooping activity.
<a href="#">show debugging</a>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug lacp

Use the **debug lacp** privileged EXEC command to enable debugging of Link Aggregation Control Protocol (LACP) activity. Use the **no** form of this command to disable debugging output.

**debug lacp** [**all** | **event** | **fsm** | **misc** | **packet**]

**no debug lacp** [**all** | **event** | **fsm** | **misc** | **packet**]

## Syntax Description

<b>all</b>	(Optional) Display all LACP debug messages.
<b>event</b>	(Optional) Display LACP event debug messages.
<b>fsm</b>	(Optional) Display LACP finite state-machine debug messages.
<b>misc</b>	(Optional) Display miscellaneous LACP debug messages.
<b>packet</b>	(Optional) Display LACP packet debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(14)EA1	This command was introduced.

## Usage Guidelines

The **undebug lacp** command is the same as the **no debug lacp** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session** *switch-number* privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command** *stack-member-number* *LINE* privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
<a href="#">show lacp</a>	Displays LACP channel-group information.

# debug mac-notification

Use the **debug mac-notification** privileged EXEC command to enable debugging of MAC notification events. Use the **no** form of this command to disable debugging output.

**debug mac-notification**

**no debug mac-notification**

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

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebg mac-notification** command is the same as the **no debug mac-notification** command. When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<b>show mac address-table notification</b>	Displays the MAC address notification information for all interfaces or the specified interface.

# debug matm

Use the **debug matm** privileged EXEC command to enable debugging of platform-independent MAC address management. Use the **no** form of this command to disable debugging output.

**debug matm**

**no debug matm**

---

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

---

**Defaults** Debugging is disabled.

---

**Command Modes** Privileged EXEC

---

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

---



---

**Usage Guidelines** The **undebug matm** command is the same as the **no debug matm** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

---

Related Commands	Command	Description
	<a href="#">debug platform matm</a>	Displays information about platform-dependent MAC address management.
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

---

# debug monitor

Use the **debug monitor** privileged EXEC command to enable debugging of the Switched Port Analyzer (SPAN) feature. Use the **no** form of this command to disable debugging output.

**debug monitor** { **all** | **errors** | **idb-update** | **info** | **list** | **notifications** | **platform** | **requests** | **snmp** }

**no debug monitor** { **all** | **errors** | **idb-update** | **info** | **list** | **notifications** | **platform** | **requests** | **snmp** }

## Syntax Description

<b>all</b>	Display all SPAN debug messages.
<b>errors</b>	Display detailed SPAN error debug messages.
<b>idb-update</b>	Display SPAN interface description block (IDB) update-trace debug messages.
<b>info</b>	Display SPAN informational-tracing debug messages.
<b>list</b>	Display SPAN port and VLAN-list tracing debug messages.
<b>notifications</b>	Display SPAN notification debug messages.
<b>platform</b>	Display SPAN platform-tracing debug messages.
<b>requests</b>	Display SPAN request debug messages.
<b>snmp</b>	Display SPAN and Simple Network Management Protocol (SNMP) tracing debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebg monitor** command is the same as the **no debug monitor** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session** *switch-number* privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command** *stack-member-number* *LINE* privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## ■ debug monitor

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<b>show monitor</b>	Displays information about all SPAN and remote SPAN (RSPAN) sessions on the switch.



# debug mvrdbg

Use the **debug mvrdbg** privileged EXEC command to enable debugging of Multicast VLAN Registration (MVR). Use the **no** form of this command to disable debugging output.

```
debug mvrdbg {all | events | igmpsn | management | ports}
```

```
no debug mvrdbg {all | events | igmpsn | management | ports}
```

## Syntax Description

<b>all</b>	Display all MVR activity debug messages.
<b>events</b>	Display MVR event-handling debug messages.
<b>igmpsn</b>	Display MVR Internet Group Management Protocol (IGMP) snooping-activity debug messages.
<b>management</b>	Display MVR management-activity debug messages.
<b>ports</b>	Display MVR port debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug mvrdbg** command is the same as the **no debug mvrdbg** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
<b>show mvr</b>	Displays the current MVR configuration.

# debug nvram

Use the **debug nvram** privileged EXEC command to enable debugging of NVRAM activity. Use the **no** form of this command to disable debugging output.

**debug nvram**

**no debug nvram**

---

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

---

**Defaults** Debugging is disabled.

---

**Command Modes** Privileged EXEC

---

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

---



---

**Usage Guidelines** The **undebbug nvram** command is the same as the **no debug nvram** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

---

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

---

# debug pagp

Use the **debug pagp** privileged EXEC command to enable debugging of Port Aggregation Protocol (PAgP) activity. Use the **no** form of this command to disable debugging output.

**debug pagp** [**all** | **event** | **fsm** | **misc** | **packet**]

**no debug pagp** [**all** | **event** | **fsm** | **misc** | **packet**]

## Syntax Description

<b>all</b>	(Optional) Display all PAgP debug messages.
<b>event</b>	(Optional) Display PAgP event debug messages.
<b>fsm</b>	(Optional) Display PAgP finite state-machine debug messages.
<b>misc</b>	(Optional) Display miscellaneous PAgP debug messages.
<b>packet</b>	(Optional) Display PAgP packet debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug pagp** command is the same as the **no debug pagp** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session** *switch-number* privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command** *stack-member-number* *LINE* privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
<b>show pagp</b>	Displays PAgP channel-group information.

# debug platform acl

Use the **debug platform acl** privileged EXEC command to enable debugging of the access control list (ACL) manager. Use the **no** form of this command to disable debugging output.

```
debug platform acl {all | exit | label | main | racl | stack | vacl | vmap | warn}
```

```
no debug platform acl {all | exit | label | main | racl | stack | vacl | vmap | warn}
```

## Syntax Description

<b>all</b>	Display all ACL manager debug messages.
<b>exit</b>	Display ACL exit-related debug messages.
<b>label</b>	Display ACL label-related debug messages.
<b>main</b>	Display the main or important ACL debug messages.
<b>racl</b>	Display router ACL related debug messages.
<b>stack</b>	Display ACL stack-related debug messages.
<b>vacl</b>	Display VLAN ACL-related debug messages.
<b>vmap</b>	Display ACL VLAN-map-related debug messages.
<b>warn</b>	Display ACL warning-related debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform acl** command is the same as the **no debug platform acl** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform cli-redirection main

Use the **debug platform cli-redirection main** privileged EXEC command to enable debugging of the main (important) command-line interface (CLI) redirection events. Use the **no** form of this command to disable debugging output.

**debug platform cli-redirection main**

**no debug platform cli-redirection main**

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

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebug platform cli-redirection main** command is the same as the **no debug platform cli-redirection main** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform configuration

Use the **debug platform configuration** privileged EXEC command to enable debugging of configuration file activity across the stack. Use the **no** form of this command to disable debugging output.

**debug platform configuration** {all | reception | transmission}

**no debug platform configuration** {all | reception | transmission}

## Syntax Description

<b>all</b>	Display debug messages for all configuration file transmission and reception events throughout the stack.
<b>reception</b>	Display debug messages for configuration file reception from other stack members.
<b>transmission</b>	Display debug messages for configuration file transmission to other stack members.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform configuration** command is the same as the **no debug platform configuration** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

## debug platform cpu-queues

Use the **debug platform cpu-queues** privileged EXEC command to enable debugging of platform central processing unit (CPU) receive queues. Use the **no** form of this command to disable debugging output.

```
debug platform cpu-queues { broadcast-q | cbt-to-spt-q | cpuhub-q | host-q | icmp-q |
  igmp-snooping-q | layer2-protocol-q | logging-q | remote-console-q | routing-protocol-q |
  rpffail-q | software-fwd-q | stp-q }
```

```
no debug platform cpu-queues { broadcast-q | cbt-to-spt-q | cpuhub-q | host-q | icmp-q |
  igmp-snooping-q | layer2-protocol-q | logging-q | remote-console-q | routing-protocol-q |
  rpffail-q | software-fwd-q | stp-q }
```

Syntax Description		
<b>broadcast-q</b>	Display debug messages about packets received by the broadcast queue.	
<b>cbt-to-spt-q</b>	Display debug messages about packets received by the core-based tree to shortest-path tree (cbt-to-spt) queue.	
<b>cpuhub-q</b>	Display debug messages about packets received by the CPU heartbeat queue.	
<b>host-q</b>	Display debug messages about packets received by the host queue.	
<b>icmp-q</b>	Display debug messages about packets received by the Internet Control Message Protocol (ICMP) queue.	
<b>igmp-snooping-q</b>	Display debug messages about packets received by the Internet Group Management Protocol (IGMP)-snooping queue.	
<b>layer2-protocol-q</b>	Display debug messages about packets received by the Layer 2 protocol queue.	
<b>logging-q</b>	Display debug messages about packets received by the logging queue.	
<b>remote-console-q</b>	Display debug messages about packets received by the remote console queue.	
<b>routing-protocol-q</b>	Display debug messages about packets received by the routing protocol queue.	
<b>rpffail-q</b>	Display debug messages about packets received by the reverse path forwarding (RFP) failure queue.	
<b>software-fwd-q</b>	Debug packets received by the software forwarding queue.	
<b>stp-q</b>	Debug packets received by the Spanning Tree Protocol (STP) queue.	

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines**

The **undebg platform cpu-queues** command is the same as the **no debug platform cpu-queues** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

**Related Commands**

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select



# debug platform device-manager

Use the **debug platform device-manager** privileged EXEC command to enable debugging of the platform-dependent device manager. Use the **no** form of this command to disable debugging output.

```
debug platform device-manager {all | device-info | poll | port-download | trace}
```

```
no debug platform device-manager {all | device-info | poll | port-download | trace}
```

## Syntax Description

<b>all</b>	Display all platform device manager debug messages.
<b>device-info</b>	Display platform device manager device structure debug messages.
<b>poll</b>	Display platform device manager 1-second poll debug messages.
<b>port-download</b>	Display platform device manager remote procedure call (RPC) usage debug messages.
<b>trace</b>	Trace platform device manager function entry and exit debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebg platform device-manager** command is the same as the **no debug platform device-manager** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform dot1x

Use the **debug platform dot1x** privileged EXEC command to enable debugging of stack-related 802.1x events. Use the **no** form of this command to disable debugging output.

**debug platform dot1x** { **initialization** | **interface-configuration** | **rpc** }

**no debug platform dot1x** { **initialization** | **interface-configuration** | **rpc** }

## Syntax Description

<b>initialization</b>	Display 802.1x initialization sequence debug messages.
<b>interface-configuration</b>	Display 802.1x interface configuration-related debug messages.
<b>rpc</b>	Display 802.1x remote procedure call (RPC) request debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform dot1x** command is the same as the **no debug platform dot1x** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform etherchannel

Use the **debug platform etherchannel** privileged EXEC command to enable debugging of platform-dependent EtherChannel events. Use the **no** form of this command to disable debugging output.

```
debug platform etherchannel {init | link-up | rpc-detailed | rpc-generic | warnings}
```

```
no debug platform etherchannel {init | link-up | rpc-detailed | rpc-generic | warnings}
```

## Syntax Description

<b>init</b>	Display EtherChannel module initialization debug messages.
<b>link-up</b>	Display EtherChannel link-up and link-down related debug messages.
<b>rpc-detailed</b>	Display detailed EtherChannel remote procedure call (RPC) debug messages.
<b>rpc-generic</b>	Display EtherChannel RPC generic debug messages.
<b>warnings</b>	Display EtherChannel warning debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform etherchannel** command is the same as the **no debug platform etherchannel** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform fallback-bridging

Use the **debug platform fallback-bridging** privileged EXEC command to enable debugging of the platform-dependent fallback bridging manager. Use the **no** form of this command to disable debugging output.

```
debug platform fallback-bridging [error | retry | rpc {events | messages}]
```

```
no debug platform fallback-bridging [error | retry | rpc {events | messages}]
```

## Syntax Description

<b>error</b>	(Optional) Display fallback bridging manager error condition messages.
<b>retry</b>	(Optional) Display fallback bridging manager retry messages.
<b>rpc {events   messages}</b>	(Optional) Display fallback bridging debugging information. The keywords have these meanings: <ul style="list-style-type: none"> <li><b>events</b>—Display remote procedure call (RPC) events.</li> <li><b>messages</b>—Display RPC messages.</li> </ul>

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

If you do not specify a keyword, all fallback bridging manager debug messages appear.

The **undebug platform fallback-bridging** command is the same as the **no debug platform fallback-bridging** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform forw-tcam

Use the **debug platform forw-tcam** privileged EXEC command to enable debugging of the forwarding ternary content addressable memory (TCAM) manager. Use the **no** form of this command to disable debugging output.

**debug platform forw-tcam** [**adjustment** | **allocate** | **audit** | **error** | **move** | **read** | **write**]

**no debug platform forw-tcam** [**adjustment** | **allocate** | **audit** | **error** | **move** | **read** | **write**]

## Syntax Description

<b>adjustment</b>	(Optional) Display TCAM manager adjustment debug messages.
<b>allocate</b>	(Optional) Display TCAM manager allocation debug messages.
<b>audit</b>	(Optional) Display TCAM manager audit messages.
<b>error</b>	(Optional) Display TCAM manager error messages.
<b>move</b>	(Optional) Display TCAM manager move messages.
<b>read</b>	(Optional) Display TCAM manager read messages.
<b>write</b>	(Optional) Display TCAM manager write messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

If you do not specify a keyword, all forwarding TCAM manager debug messages appear.

The **undebg platform forw-tcam** command is the same as the **no debug platform forw-tcam** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

**■** debug platform forw-tcam

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform ip arp inspection

Use the **debug platform ip arp inspection** privileged EXEC command to debug dynamic Address Resolution Protocol (ARP) inspection events. Use the **no** form of this command to disable debugging output.

**debug platform ip arp inspection {all | error | event | packet | rpc}**

**no debug platform ip arp inspection {all | error | event | packet | rpc}**

This command is available only if your switch is running the enhanced multilayer image (EMI).

## Syntax Description

<b>all</b>	Display all dynamic ARP inspection debug messages.
<b>error</b>	Display dynamic ARP inspection error debug messages.
<b>event</b>	Display dynamic ARP inspection event debug messages.
<b>packet</b>	Display dynamic ARP inspection packet-related debug messages.
<b>rpc</b>	Display dynamic ARP inspection remote procedure call (RPC) request debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.2(20)SE	This command was introduced.

## Usage Guidelines

The **undebg platform ip arp inspection** command is the same as the **no debug platform ip arp inspection** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<a href="#">show ip arp inspection</a>	Displays the dynamic ARP inspection configuration and operating state.
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform ip dhcp

Use the **debug platform ip dhcp** privileged EXEC command to debug DHCP events. Use the **no** form of this command to disable debugging output.

**debug platform ip dhcp** [**all** | **error** | **event** | **packet** | **rpc**]

**no debug platform ip dhcp** [**all** | **error** | **event** | **packet** | **rpc**]

## Syntax Description

<b>all</b>	(Optional) Display all DHCP debug messages.
<b>error</b>	(Optional) Display DHCP error debug messages.
<b>event</b>	(Optional) Display DHCP event debug messages.
<b>packet</b>	(Optional) Display DHCP packet-related debug messages.
<b>rpc</b>	(Optional) Display DHCP remote procedure call(RPC) request debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(19)EA1	This command was introduced.

## Usage Guidelines

The **undebug platform ip dhcp** command is the same as the **no debug platform ip dhcp** command. When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<a href="#">show ip dhcp snooping</a>	Displays the DHCP snooping configuration.
<a href="#">show ip dhcp snooping binding</a>	Displays the DHCP snooping binding information.
<a href="#">show debugging</a>	Displays information about the types of debugging that are enabled. For syntax information, select



## debug platform ip igmp snooping

Use the **debug platform ip igmp snooping** privileged EXEC command to enable debugging of platform-dependent Internet Group Management Protocol (IGMP) snooping. Use the **no** form of this command to disable debugging output.

```
debug platform ip igmp snooping {all | di | error | event | group | mgmt | pak | retry | rpc | warn}
```

```
debug platform ip igmp snooping pak {ip-address | error | ipopt | leave | query | report | rx | svi | tx}
```

```
debug platform ip igmp snooping rpc [cfg | l3mm | misc | vlan]
```

```
no debug platform ip igmp snooping {all | di | error | event | group | mgmt | pak | retry | rpc | warn}
```

### Syntax Description

<b>all</b>	Display all IGMP snooping debug messages.
<b>di</b>	Display IGMP snooping destination index (di) coordination remote procedure call (RPC) debug messages.
<b>error</b>	Display IGMP snooping error messages.
<b>event</b>	Display IGMP snooping event debug messages.
<b>group</b>	Display IGMP snooping group debug messages.
<b>mgmt</b>	Display IGMP snooping management debug messages.
<b>pak</b> { <i>ip-address</i>   <b>error</b>   <b>ipopt</b>   <b>leave</b>   <b>query</b>   <b>report</b>   <b>rx</b>   <b>svi</b>   <b>tx</b> }	<p>Display IGMP snooping packet event debug messages. The keywords have these meanings:</p> <ul style="list-style-type: none"> <li>• <i>ip-address</i>—IP address of the IGMP group.</li> <li>• <b>error</b>—Display IGMP snooping packet error debug messages.</li> <li>• <b>ipopt</b>—Display IGMP snooping IP bridging options debug messages.</li> <li>• <b>leave</b>—Display IGMP snooping leave debug messages.</li> <li>• <b>query</b>—Display IGMP snooping query debug messages.</li> <li>• <b>report</b>—Display IGMP snooping report debug messages.</li> <li>• <b>rx</b>—Display IGMP snooping received packet debug messages.</li> <li>• <b>svi</b>—Display IGMP snooping switched virtual interface (SVI) packet debug messages.</li> <li>• <b>tx</b>—Display IGMP snooping sent packet debug messages.</li> </ul>
<b>retry</b>	Display IGMP snooping retry debug messages.

## debug platform ip igmp snooping

<b>rpc</b> [ <b>cfg</b>   <b>l3mm</b>   <b>misc</b>   <b>vlan</b> ]	<p>Display IGMP snooping remote procedure call (RPC) event debug messages. The keywords have these meanings:</p> <ul style="list-style-type: none"> <li>• <b>cfg</b>—(Optional) Display IGMP snooping RPC debug messages.</li> <li>• <b>l3mm</b>—(Optional) IGMP snooping Layer 3 multicast router group RPC debug messages.</li> <li>• <b>misc</b>—(Optional) IGMP snooping miscellaneous RPC debug messages.</li> <li>• <b>vlan</b>—(Optional) IGMP snooping VLAN assert RPC debug messages.</li> </ul>
<b>warn</b>	Display IGMP snooping warning messages.

### Defaults

Debugging is disabled.

### Command Modes

Privileged EXEC

### Command History

Release	Modification
12.1(11)AX	This command was introduced.

### Usage Guidelines

The **undebg platform ip igmp snooping** command is the same as the **no debug platform ip igmp snooping** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

### Related Commands

Command	Description
<a href="#">debug ip igmp snooping</a>	Displays information about platform-independent IGMP snooping activity.
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform ip multicast

Use the **debug platform ip multicast** privileged EXEC command to enable debugging of IP multicast routing. Use the **no** form of this command to disable debugging output.

**debug platform ip multicast** { **all** | **mdb** | **mdfs-rp-retry** | **midb** | **mroute-rp** | **resources** | **retry** | **rpf-throttle** | **snoop-events** | **software-forward** | **swidb-events** | **vlan-locks** }

**no debug platform ip multicast** { **all** | **mdb** | **mdfs-rp-retry** | **midb** | **mroute-rp** | **resources** | **retry** | **rpf-throttle** | **snoop-events** | **software-forward** | **swidb-events** | **vlan-locks** }

Syntax Description		
<b>all</b>	Display all platform IP-multicast event debug messages.	<b>Note</b> Using this command can degrade the performance of the switch.
<b>mdb</b>	Display IP-multicast debug messages for multicast distributed fast switching (MDFS) multicast descriptor block (mdb) events.	
<b>mdfs-rp-retry</b>	Display IP-multicast MDFS rendezvous point (RP) retry event debug messages.	
<b>midb</b>	Display IP-multicast MDFS multicast interface descriptor block (MIDB) debug messages.	
<b>mroute-rp</b>	Display IP-multicast RP event debug messages.	
<b>resources</b>	Display IP-multicast hardware resource debug messages.	
<b>retry</b>	Display IP-multicast retry processing event debug messages.	
<b>rpf-throttle</b>	Display IP-multicast reverse path forwarding (RPF) throttle event debug messages.	
<b>snoop-events</b>	Display IP-multicast IGMP snooping event debug messages.	
<b>software-forward</b>	Display IP-multicast software forwarding event debug messages.	
<b>swidb-events</b>	Display IP-multicast MDFS software interface descriptor block (swidb) or global event debug messages.	
<b>vlan-locks</b>	Display IP-multicast VLAN lock and unlock event debug messages.	

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines**

The **undebug platform ip multicast** command is the same as the **no debug platform ip multicast** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session *switch-number*** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command *stack-member-number LINE*** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

**Related Commands**

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform ip unicast

Use the **debug platform ip unicast** privileged EXEC command to enable debugging of platform-dependent IP unicast routing. Use the **no** form of this command to disable debugging output.

```
debug platform ip unicast {adjacency | all | arp | dhcp | errors | events | interface | mpath |
registries | retry | route | rpc | standby | statistics}
```

```
no debug platform ip unicast {adjacency | all | arp | dhcp | errors | events | interface | mpath |
registries | retry | route | rpc | standby | statistics}
```

Syntax Description	
<b>adjacency</b>	Display IP unicast routing adjacency programming event debug messages.
<b>all</b>	Display all platform IP unicast routing debug messages. <b>Note</b> Using this command can degrade the performance of the switch.
<b>arp</b>	Display IP unicast routing Address Resolution Protocol (ARP) and ARP throttling debug messages.
<b>dhcp</b>	Display IP unicast routing DHCP dynamic address-related event debug messages.
<b>errors</b>	Display all IP unicast routing error debug messages, including resource allocation failures.
<b>events</b>	Display all IP unicast routing event debug messages, including registry and miscellaneous events.
<b>interface</b>	Display IP unicast routing interface event debug messages.
<b>mpath</b>	Display IP unicast routing multi-path adjacency programming event debug messages (present when performing equal or unequal cost routing).
<b>registries</b>	Display IP unicast routing forwarding information database (FIB), adjacency add, update, and delete registry event debug messages.
<b>retry</b>	Display IP unicast routing reprogram FIBs with ternary content addressable memory (TCAM) allocation failure debug messages.
<b>route</b>	Display IP unicast routing FIB TCAM programming event debug messages.
<b>rpc</b>	Display IP unicast routing Layer 3 unicast remote procedure call (RPC) interaction debug messages.
<b>standby</b>	Display IP unicast routing standby event debug messages, helpful in troubleshooting Hot Standby Routing Protocol (HSRP) issues.
<b>statistics</b>	Display IP unicast routing statistics gathering-related event debug messages.

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines**

The **undebug platform ip unicast** command is the same as the **no debug platform ip unicast** command. When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session** *switch-number* privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command** *stack-member-number* *LINE* privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

**Related Commands**

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform ipc

Use the **debug platform ipc** privileged EXEC command to enable debugging of the platform-dependent Interprocess Communication (IPC) Protocol. Use the **no** form of this command to disable debugging output.

```
debug platform ipc {all | init | receive | send | trace}
```

```
no debug platform {all | init | receive | send | trace}
```

## Syntax Description

<b>all</b>	Display all platform IPC debug messages. <b>Note</b> Using this command can degrade the performance of the switch.
<b>init</b>	Display debug messages related to IPC initialization.
<b>receive</b>	Display IPC traces each time an IPC packet is received by the switch.
<b>send</b>	Display IPC traces each time an IPC packet is sent by the switch.
<b>trace</b>	Display IPC trace debug messages, tracing the code path as the IPC functions are executed.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform ipc** command is the same as the **no debug platform ipc**.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform led

Use the **debug platform led** privileged EXEC command to enable debugging of light-emitting diode (LED) actions. Use the **no** form of this command to disable debugging output.

```
debug platform led {generic | signal | stack}
```

```
no debug platform led {generic | signal | stack}
```

## Syntax Description

<b>generic</b>	Display LED generic action debug messages.
<b>signal</b>	Display LED signal bit map debug messages.
<b>stack</b>	Display LED stack action debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform led** command is the same as the **no debug platform led** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select



# debug platform matm

Use the **debug platform matm** privileged EXEC command to enable debugging of platform-dependent MAC address management. Use the **no** form of this command to disable debugging output.

```
debug platform matm { aging | all | ec-aging | errors | learning | rpc | secure-address | warnings }
```

```
no debug platform matm { aging | all | ec-aging | errors | learning | rpc | secure-address | warnings }
```

## Syntax Description

<b>aging</b>	Display MAC address aging debug messages.
<b>all</b>	Display all platform MAC address management event debug messages.
<b>ec-aging</b>	Display EtherChannel address aging-related debug messages.
<b>errors</b>	Display MAC address management error messages.
<b>learning</b>	Display MAC address management address-learning debug messages.
<b>rpc</b>	Display MAC address management remote procedure call (RPC) related debug messages.
<b>secure-address</b>	Display MAC address management secure address learning debug messages.
<b>warning</b>	Display MAC address management warning messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform matm** command is the same as the **no debug platform matm** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

**debug platform matm**

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<a href="#">debug matm</a>	Displays information about platform-independent MAC address management.
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

# debug platform messaging application

Use the **debug platform messaging application** privileged EXEC command to enable debugging of application messaging activity. Use the **no** form of this command to disable debugging output.

```
debug platform messaging application {all | badpak | cleanup | events | memerr | messages |
stackchg | usererr}
```

```
no debug platform messaging application {all | badpak | cleanup | events | memerr | messages
| stackchg | usererr}
```

Syntax Description	all	Display all application-messaging debug messages.
	<b>badpak</b>	Display bad-packet debug messages.
	<b>cleanup</b>	Display clean-up debug messages.
	<b>events</b>	Display event debug messages.
	<b>memerr</b>	Display memory-error debug messages.
	<b>messages</b>	Display application-messaging debug messages.
	<b>stackchg</b>	Display stack-change debug messages.
	<b>usererr</b>	Display user-error debug messages.

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebug platform messaging application** command is the same as the **no debug platform messaging application** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug platform phy

Use the **debug platform phy** privileged EXEC command to enable debugging of PHY driver information. Use the **no** form of this command to disable debugging output.

```
debug platform phy {automdix | flowcontrol | forced | init-seq | link-status | read | sfp |
show-controller | speed | write}
```

```
no debug platform phy {automdix | flowcontrol | forced | init-seq | link-status | read | sfp |
show-controller | speed | write}
```

## Syntax Description

<b>automdix</b>	Display PHY automatic medium-dependent interface crossover (Auto-MDIX) debug messages.
<b>flowcontrol</b>	Display PHY flowcontrol debug messages.
<b>forced</b>	Display PHY forced-mode debug messages.
<b>init-seq</b>	Display PHY initialization-sequence debug messages.
<b>link-status</b>	Display PHY link-status debug messages.
<b>read</b>	Display PHY-read debug messages.
<b>sfp</b>	Display PHY small form-factor pluggable (SFP) modules debug messages.
<b>show-controller</b>	Display PHY show-controller debug messages.
<b>speed</b>	Display PHY speed-change debug messages.
<b>write</b>	Display PHY-write debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.
12.1(14)EA1	The <b>automdix</b> keyword was added.

## Usage Guidelines

The **undebug platform phy** command is the same as the **no debug platform phy** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	show debugging	Displays information about the types of debugging that are enabled. For syntax information, refer to the

## debug platform pm

Use the **debug platform pm** privileged EXEC command to enable debugging of the platform-dependent port manager software module. Use the **no** form of this command to disable debugging output.

```
debug platform pm {all | counters | errdisable | etherchnl | exceptions | hpm-events | idb-events
| if-numbers | ios-events | link-status | platform | pm-events | pm-vectors [detail] | rpc
[general | oper-info | state | vectors | vp-events] | soutput | stack-manager | sync | vlans }
```

```
no debug platform pm {all | counters | errdisable | etherchnl | exceptions | hpm-events |
idb-events | if-numbers | ios-events | link-status | platform | pm-events | pm-vectors [detail]
| rpc [general | oper-info | state | vectors | vp-events] | soutput | stack-manager | sync | vlans }
```

### Syntax Description

<b>all</b>	Display all port-manager debug messages.
<b>counters</b>	Display counters for remote procedure call (RPC) debug messages.
<b>errdisable</b>	Display error-disabled related-events debug messages.
<b>etherchnl</b>	Display EtherChannel related-events debug messages.
<b>exceptions</b>	Display system exception debug messages.
<b>hpm-events</b>	Display platform port-manager event debug messages.
<b>idb-events</b>	Display interface descriptor block (IDB) related-events debug messages.
<b>if-numbers</b>	Display interface-number translation-event debug messages.
<b>ios-events</b>	Display IOS event debug messages.
<b>link-status</b>	Display interface link-detection event debug messages.
<b>platform</b>	Display port-manager function-event debug messages.
<b>pm-events</b>	Display port manager event debug messages.
<b>pm-vectors [detail]</b>	Display port-manager vector-related-event debug messages. The keyword has this meaning: <ul style="list-style-type: none"> <li><b>detail</b>—Display vector-function details.</li> </ul>
<b>rpc [general   oper-info   state   vectors   vp-events]</b>	Display RPC related-event debug messages. The keywords have these meanings: <ul style="list-style-type: none"> <li><b>general</b>—(Optional) Display RPC general events.</li> <li><b>oper-info</b>—(Optional) Display operational- and informational-related RPC messages.</li> <li><b>state</b>—(Optional) Display administrative- and operational-related RPC messages.</li> <li><b>vectors</b>—(Optional) Display vector-related RPC messages.</li> <li><b>vp-events</b>—(Optional) Display virtual ports related-events RP messages.</li> </ul>
<b>soutput</b>	Display IDB output vector event debug messages.
<b>stack-manager</b>	Display stack-manager related-events debug messages.
<b>sync</b>	Display operational synchronization and VLAN line-state event debug messages.
<b>vlans</b>	Display VLAN creation and deletion event debug messages.

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebg platform pm** command is the same as the **no debug platform pm** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session *switch-number*** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command *stack-member-number LINE*** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug platform port-asic

Use the **debug platform port-asic** privileged EXEC command to enable debugging of the port application-specific integrated circuit (ASIC) driver. Use the **no** form of this command to disable debugging output.

```
debug platform port-asic {interrupt | periodic | read | stack | write}
```

```
no debug platform port-asic {interrupt | periodic | read | stack | write}
```

## Syntax Description

<b>interrupt</b>	Display port-ASIC interrupt-related function debug messages.
<b>periodic</b>	Display port-ASIC periodic-function-call debug messages.
<b>read</b>	Display port-ASIC read debug messages.
<b>stack</b>	Display stacking-related function debug messages.
<b>write</b>	Display port-ASIC write debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform port-asic** command is the same as the **no debug platform port-asic** command. When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the



# debug platform port-security

Use the **debug platform port-security** privileged EXEC command to enable debugging of platform-dependent port-security information. Use the **no** form of this command to disable debugging output.

**debug platform port-security** {**add** | **aging** | **all** | **delete** | **errors** | **rpc** | **warnings**}

**no debug platform port-security** {**add** | **aging** | **all** | **delete** | **errors** | **rpc** | **warnings**}

## Syntax Description

<b>add</b>	Display secure address addition debug messages.
<b>aging</b>	Display secure address aging debug messages.
<b>all</b>	Display all port-security debug messages.
<b>delete</b>	Display secure address deletion debug messages.
<b>errors</b>	Display port-security error debug messages.
<b>rpc</b>	Display remote procedure call (RPC) debug messages.
<b>warnings</b>	Display warning debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform port-security** command is the same as the **no debug platform port-security** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug platform qos-acl-tcam

Use the **debug platform qos-acl-tcam** privileged EXEC command to enable debugging of the quality of service (QoS) and access control list (ACL) ternary content addressable memory (TCAM) manager software. Use the **no** form of this command to disable debugging output.

```
debug platform qos-acl-tcam {all | ctcam | errors | labels | mask | rpc | tcam}
```

```
no debug platform qos-acl-tcam {all | ctcam | errors | labels | mask | rpc | tcam}
```

## Syntax Description

<b>all</b>	Display all QoS and ACL TCAM (QATM) manager debug messages.
<b>ctcam</b>	Display Cisco TCAM (CTCAM) related-events debug messages.
<b>errors</b>	Display QATM error-related-events debug messages.
<b>labels</b>	Display QATM label-related-events debug messages.
<b>mask</b>	Display QATM mask-related-events debug messages.
<b>rpc</b>	Display QATM remote procedure call (RPC) related-events debug messages.
<b>tcam</b>	Display QATM TCAM-related events debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform qos-acl-tcam** command is the same as the **no debug platform qos-acl-tcam** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug platform remote-commands

Use the **debug platform remote-commands** privileged EXEC command to enable debugging of remote commands. Use the **no** form of this command to disable debugging output.

**debug platform remote-commands**

**no debug platform remote-commands**

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

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebg platform remote-commands** command is the same as the **no debug platform remote-commands** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug platform resource-manager

Use the **debug platform resource-manager** privileged EXEC command to enable debugging of the resource manager software. Use the **no** form of this command to disable debugging output.

```
debug platform resource-manager {all | dm | erd | errors | madmed | sd | stats | vld}
```

```
no debug platform resource-manager {all | dm | erd | errors | madmed | sd | stats | vld}
```

## Syntax Description

<b>all</b>	Display all resource manager debug messages.
<b>dm</b>	Display destination-map debug messages.
<b>erd</b>	Display equal-cost-route descriptor-table debug messages.
<b>errors</b>	Display error debug messages.
<b>madmed</b>	Display the MAC address descriptor table and multi-expansion descriptor table debug messages.
<b>sd</b>	Display the station descriptor table debug messages.
<b>stats</b>	Display statistics debug messages.
<b>vld</b>	Display the VLAN-list descriptor debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform resource-manager** command is the same as the **no debug platform resource-manager** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug platform snmp

Use the **debug platform snmp** privileged EXEC command to enable debugging of the platform-dependent Simple Network Management Protocol (SNMP) software. Use the **no** form of this command to disable debugging output.

**debug platform snmp**

**no debug platform snmp**

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

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebg platform snmp** command is the same as the **no debug platform snmp** command. When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug platform span

Use the **debug platform span** privileged EXEC command to enable debugging of the platform-dependent Switched Port Analyzer (SPAN) software. Use the **no** form of this command to disable debugging output.

**debug platform span**

**no debug platform span**

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

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebbug platform span** command is the same as the **no debug platform span** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug platform stack-manager

Use the **debug platform stack-manager** privileged EXEC command to enable debugging of the stack manager software. Use the **no** form of this command to disable debugging output.

```
debug platform stack-manager {all | rpc | sdp | sim | ssm | trace}
```

```
no debug platform stack-manager {all | rpc | sdp | sim | ssm | trace}
```

## Syntax Description

<b>all</b>	Display all stack manager debug messages.
<b>rpc</b>	Display stack manager remote procedure call (RPC) usage debug messages.
<b>sdp</b>	Display the Stack Discovery Protocol (SDP) debug messages.
<b>sim</b>	Display the stack information module debug messages.
<b>ssm</b>	Display the stack state-machine debug messages.
<b>trace</b>	Trace the stack manager entry and exit debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform stack-manager** command is the same as the **no debug platform stack-manager** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug platform supervisor-asic

Use the **debug platform supervisor-asic** privileged EXEC command to enable debugging of the supervisor application-specific integrated circuit (ASIC). Use the **no** form of this command to disable debugging output.

```
debug platform supervisor-asic {all | errors | receive | send}
```

```
no debug platform supervisor-asic {all | errors | receive | send}
```

## Syntax Description

<b>all</b>	Display all supervisor-ASIC event debug messages.
<b>errors</b>	Display the supervisor-ASIC error debug messages.
<b>receive</b>	Display the supervisor-ASIC receive debug messages.
<b>send</b>	Display the supervisor-ASIC send debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebg platform supervisor-asic** command is the same as the **no debug platform supervisor-asic** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the



# debug platform sw-bridge

Use the **debug platform sw-bridge** privileged EXEC command to enable debugging of the software bridging function. Use the **no** form of this command to disable debugging output.

```
debug platform sw-bridge {broadcast | control | multicast | packet | unicast}
```

```
no debug platform sw-bridge {broadcast | control | multicast | packet | unicast}
```

## Syntax Description

<b>broadcast</b>	Display broadcast-data debug messages.
<b>control</b>	Display protocol-packet debug messages.
<b>multicast</b>	Display multicast-data debug messages.
<b>packet</b>	Display sent and received data debug messages.
<b>unicast</b>	Display unicast-data debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform sw-bridge** command is the same as the **no debug platform sw-bridge** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

## debug platform tcam

Use the **debug platform tcam** privileged EXEC command to enable debugging of ternary content addressable memory (TCAM) access and lookups. Use the **no** form of this command to disable debugging output.

```
debug platform tcam {log | read | search | write}
```

```
debug platform tcam log l2 {acl {input | output} | local | qos}
```

```
debug platform tcam log l3 {acl {input | output} | ipv6 {acl {input | output} | local | qos |  
secondary} | local | qos | secondary}
```

```
debug platform tcam read {reg | ssram | tcam}
```

```
debug platform tcam search
```

```
debug platform tcam write {forw-ram | reg | tcam}
```

```
no debug platform tcam {log | read | search | write}
```

```
no debug platform tcam log l2 {acl {input | output} | local | qos}
```

```
no debug platform tcam log l3 {acl {input | output} | ipv6 {acl {input | output} | local | qos |  
secondary} | local | qos | secondary}
```

```
no debug platform tcam read {reg | ssram | tcam}
```

```
no debug platform tcam search
```

```
no debug platform tcam write {forw-ram | reg | tcam}
```

<b>Syntax Description</b>	<p><b>log l2 {acl {input   output}   local   qos}</b> Display Layer 2 field-based CAM look-up type debug messages. The keywords have these meanings:</p> <ul style="list-style-type: none"> <li>• <b>acl {input   output}</b>—Display input or output ACL look-up debug messages.</li> <li>• <b>local</b>—Display local forwarding look-up debug messages.</li> <li>• <b>qos</b>—Display classification and quality of service (QoS) look-up debug messages.</li> </ul> <hr/> <p><b>l3 {acl {input   output}   ipv6 {acl {input   output}   local   qos   secondary}   local   qos   secondary}</b> Display Layer 3 field-based CAM look-up type debug messages. The keywords have these meanings:</p> <ul style="list-style-type: none"> <li>• <b>acl {input   output}</b>—Display input or output ACL look-up debug messages.</li> <li>• <b>ipv6 {acl {input   output}   local   qos   secondary}</b>—Display IPv6-based look-up debug messages. Options include displaying input or output ACL look-up, local forwarding look-up, classification and QoS look-up, or secondary forwarding look-up debug messages.</li> <li>• <b>local</b>—Display local forwarding look-up debug messages.</li> <li>• <b>qos</b>—Display classification and quality of service (QoS) look-up debug messages.</li> <li>• <b>secondary</b>—Display secondary forwarding look-up debug messages.</li> </ul> <hr/> <p><b>read {reg   ssram   tcam}</b> Display TCAM-read debug messages. The keywords have these meanings:</p> <ul style="list-style-type: none"> <li>• <b>reg</b>—Display TCAM-register read debug messages.</li> <li>• <b>ssram</b>—Display synchronous static RAM (SSRAM)-read debug messages.</li> <li>• <b>tcam</b>—Display TCAM-read debug messages.</li> </ul> <hr/> <p><b>search</b> Display supervisor-initiated TCAM-search results debug messages.</p> <hr/> <p><b>write {forw-ram   reg   tcam}</b> Display TCAM-write debug messages. The keywords have these meanings:</p> <ul style="list-style-type: none"> <li>• <b>forw-ram</b>—Display forwarding-RAM write debug messages.</li> <li>• <b>reg</b>—Display TCAM-register write debug messages.</li> <li>• <b>tcam</b>—Display TCAM-write debug messages.</li> </ul>
---------------------------	--

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines**

The **undebg platform tcam** command is the same as the **no debug platform tcam** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session *switch-number*** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command *stack-member-number LINE*** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

**Related Commands**

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug platform uddl

Use the **debug platform uddl** privileged EXEC command to enable debugging of the platform-dependent UniDirectional Link Detection (UDLD) software. Use the **no** form of this command to disable debugging output.

```
debug platform uddl [all | error | rpc {events | messages}]
```

```
no debug platform uddl [all | error | rpc {events | messages}]
```

## Syntax Description

<b>all</b>	(Optional) Display all UDLD debug messages.
<b>error</b>	(Optional) Display error condition debug messages.
<b>rpc {events   messages}</b>	(Optional) Display UDLD remote procedure call (RPC) debug messages. The keywords have these meanings: <ul style="list-style-type: none"> <li><b>events</b>—Display UDLD RPC events.</li> <li><b>messages</b>—Display UDLD RPC messages.</li> </ul>

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug platform uddl** command is the same as the **no debug platform uddl** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the

# debug platform vlan

Use the **debug platform vlan** privileged EXEC command to enable debugging of the VLAN manager software. Use the **no** form of this command to disable debugging output.

```
debug platform vlan {errors | mvid | rpc}
```

```
no debug platform vlan {errors | mvid | rpc}
```

## Syntax Description

<b>errors</b>	Display VLAN error debug messages.
<b>mvid</b>	Display mapped VLAN ID allocations and free debug messages.
<b>rpc</b>	Display remote procedure call (RPC) debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebg platform vlan** command is the same as the **no debug platform vlan** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug pm

Use the **debug pm** privileged EXEC command to enable debugging of port manager (PM) activity. The port manager is a state machine that controls all the logical and physical interfaces. All features, such as VLANs, UniDirectional Link Detection (UDLD), and so forth, work with the port manager to provide switch functions. Use the **no** form of this command to disable debugging output.

```
debug pm {all | assert | card | cookies | etherchnl | messages | port | registry | sm | span | split |
          vlan | vp}
```

```
no debug pm {all | assert | card | cookies | etherchnl | messages | port | registry | sm | span | split |
            vlan | vp}
```

## Syntax Description

<b>all</b>	Display all PM debug messages.
<b>assert</b>	Display assert debug messages.
<b>card</b>	Display line-card related-events debug messages.
<b>cookies</b>	Display internal PM cookie validation debug messages.
<b>etherchnl</b>	Display EtherChannel related-events debug messages.
<b>htable</b>	Display Host Access Table events debug messages.
<b>messages</b>	Display PM debug messages.
<b>port</b>	Display port related-events debug messages.
<b>registry</b>	Display PM registry invocation debug messages.
<b>sm</b>	Display state-machine related-events debug messages.
<b>span</b>	Display spanning-tree related-events debug messages.
<b>split</b>	Display split-processor debug messages.
<b>vlan</b>	Display VLAN related-events debug messages.
<b>vp</b>	Display virtual port related-events debug messages.



### Note

Though visible in the command-line help strings, the **scp** and **pvlan** keywords are not supported.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.
12.1(14)EA1	The <b>htable</b> keyword was added.

---

**Usage Guidelines**

The **undebug pm** command is the same as the **no debug pm** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session** *switch-number* privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command** *stack-member-number* *LINE* privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

---

**Related Commands**

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

---



# debug port-security

Use the **debug port-security** privileged EXEC command to enable debugging of the allocation and states of the port security subsystem. Use the **no** form of this command to disable debugging output.

**debug port-security**

**no debug port-security**

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

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebg port-security** command is the same as the **no debug port-security** command. When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<b>show port-security</b>	Displays port-security settings for an interface or for the switch.

# debug qos-manager

Use the **debug qos-manager** privileged EXEC command to enable debugging of the quality of service (QoS) manager software. Use the **no** form of this command to disable debugging output.

```
debug qos-manager {all | event | verbose}
```

```
no debug qos-manager {all | event | verbose}
```

## Syntax Description

<b>all</b>	Display all QoS-manager debug messages.
<b>event</b>	Display QoS-manager related-event debug messages.
<b>verbose</b>	Display QoS-manager detailed debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug qos-manager** command is the same as the **no debug qos-manager** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, refer to the

# debug spanning-tree

Use the **debug spanning-tree** privileged EXEC command to enable debugging of spanning-tree activities. Use the **no** form of this command to disable debugging output.

```
debug spanning-tree {all | backbonefast | bpdu | bpdu-opt | config | csuf/csrt | etherchannel |
events | exceptions | general | mstp | pvst+ | root | snmp | switch | synchronization |
uplinkfast }
```

```
no debug spanning-tree {all | backbonefast | bpdu | bpdu-opt | config | csuf/csrt | etherchannel
| events | exceptions | general | mstp | pvst+ | root | snmp | switch | synchronization |
uplinkfast }
```

Syntax Description	
<b>all</b>	Display all spanning-tree debug messages.
<b>backbonefast</b>	Display BackboneFast-event debug messages.
<b>bpdu</b>	Display spanning-tree bridge protocol data unit (BPDU) debug messages.
<b>bpdu-opt</b>	Display optimized BPDU handling debug messages.
<b>config</b>	Display spanning-tree configuration change debug messages.
<b>csuf/csrt</b>	Display cross-stack UplinkFast and cross-stack rapid transition activity debug messages.
<b>etherchannel</b>	Display EtherChannel-support debug messages.
<b>events</b>	Display spanning-tree topology event debug messages.
<b>exceptions</b>	Display spanning-tree exception debug messages.
<b>general</b>	Display general spanning-tree activity debug messages.
<b>mstp</b>	Debug Multiple Spanning Tree Protocol events.
<b>pvst+</b>	Display per-VLAN spanning-tree plus (PVST+) event debug messages.
<b>root</b>	Display spanning-tree root-event debug messages.
<b>snmp</b>	Display spanning-tree Simple Network Management Protocol (SNMP) handling debug messages.
<b>synchronization</b>	Display the spanning-tree synchronization event debug messages.
<b>switch</b>	Display switch shim command debug messages. This shim is the software module that is the interface between the generic Spanning Tree Protocol (STP) code and the platform-specific code of various switch platforms.
<b>uplinkfast</b>	Display UplinkFast-event debug messages.

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

## ■ debug spanning-tree

Command History	Release	Modification
	12.1(11)AX	This command was introduced.
	12.1(14)EA1	The <b>mstp</b> and <b>csuf/csrt</b> keywords were added.

**Usage Guidelines**

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

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

**Related Commands**

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
<b>show spanning-tree</b>	Displays spanning-tree state information.

# debug spanning-tree backbonefast

Use the **debug spanning-tree backbonefast** privileged EXEC command to enable debugging of spanning-tree BackboneFast events. Use the **no** form of this command to disable debugging output.

**debug spanning-tree backbonefast** [**detail** | **exceptions**]

**no debug spanning-tree backbonefast** [**detail** | **exceptions**]

Syntax Description	detail	(Optional) Display detailed BackboneFast debug messages.
	<b>exceptions</b>	(Optional) Display spanning-tree BackboneFast-exception debug messages.

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

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

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<b>show spanning-tree</b>	Displays spanning-tree state information.

# debug spanning-tree bpdu

Use the **debug spanning-tree bpdu** privileged EXEC command to enable debugging of sent and received spanning-tree bridge protocol data units (BPDUs). Use the **no** form of this command to disable debugging output.

**debug spanning-tree bpdu** [receive | transmit]

**no debug spanning-tree bpdu** [receive | transmit]

## Syntax Description

<b>receive</b>	(Optional) Display the nonoptimized path for received BPDU debug messages.
<b>transmit</b>	(Optional) Display the nonoptimized path for sent BPDU debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

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

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
<b>show spanning-tree</b>	Displays spanning-tree state information.

# debug spanning-tree bpdu-opt

Use the **debug spanning-tree bpdu-opt** privileged EXEC command to enable debugging of optimized spanning-tree bridge protocol data units (BPDUs) handling. Use the **no** form of this command to disable debugging output.

**debug spanning-tree bpdu-opt [detail | packet]**

**no debug spanning-tree bpdu-opt [detail | packet]**

## Syntax Description

<b>detail</b>	(Optional) Display detailed optimized BPDU-handling debug messages.
<b>packet</b>	(Optional) Display packet-level optimized BPDU-handling debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

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

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

## Related Commands

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
<b>show spanning-tree</b>	Displays spanning-tree state information.

## debug spanning-tree mstp

Use the **debug spanning-tree mstp** privileged EXEC command to enable debugging of the Multiple Spanning Tree Protocol (MSTP) software. Use the **no** form of this command to disable debugging output.

```
debug spanning-tree mstp {all | boundary | bpdu-rx | bpdu-tx | errors | flush | init | migration |
  pm | proposals | region | roles | sanity_check | sync | tc | timers}
```

```
no debug spanning-tree mstp {all | boundary | bpdu-rx | bpdu-tx | errors | flush | init | migration
  | pm | proposals | region | roles | sanity_check | sync | tc | timers}
```

### Syntax Description

<b>all</b>	Enable all the debugging messages.
<b>boundary</b>	Debug flag changes at these boundaries: <ul style="list-style-type: none"> <li>An multiple spanning-tree (MST) region and a single spanning-tree region running Rapid Spanning Tree Protocol (RSTP)</li> <li>An MST region and a single spanning-tree region running 802.1D</li> <li>An MST region and another MST region with a different configuration</li> </ul>
<b>bpdu-rx</b>	Debug the received MST bridge protocol data units (BPDUs).
<b>bpdu-tx</b>	Debug the sent MST BPDUs.
<b>errors</b>	Debug MSTP errors.
<b>flush</b>	Debug the port flushing mechanism.
<b>init</b>	Debug the initialization of the MSTP data structures.
<b>migration</b>	Debug the protocol migration state machine.
<b>pm</b>	Debug MSTP port manager events.
<b>proposals</b>	Debug handshake messages between the designated switch and the root switch.
<b>region</b>	Debug the region synchronization between the switch processor (SP) and the route processor (RP).
<b>roles</b>	Debug MSTP roles.
<b>sanity_check</b>	Debug the received BPDU sanity check messages.
<b>sync</b>	Debug the port synchronization events.
<b>tc</b>	Debug topology change notification events.
<b>timers</b>	Debug the MSTP timers for start, stop, and expire events.

### Defaults

Debugging is disabled.

### Command Modes

Privileged EXEC

### Command History

Release	Modification
12.1(14)EA1	This command was introduced.



**Usage Guidelines**

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

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session** *switch-number* privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command** *stack-member-number* *LINE* privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

**Related Commands**

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
<b>show spanning-tree</b>	Displays spanning-tree state information.

## debug spanning-tree switch

Use the **debug spanning-tree switch** privileged EXEC command to enable debugging of the software interface between the Spanning Tree Protocol (STP) software module and the port manager software module. Use the **no** form of this command to disable debugging output.

```
debug spanning-tree switch {all | errors | flush | general | helper | pm | rx {decode | errors |
interrupt | process} | state | tx [decode] | uplinkfast}
```

```
no debug spanning-tree switch {all | errors | flush | general | helper | pm | rx {decode | errors |
interrupt | process} | state | tx [decode] | uplinkfast}
```

### Syntax Description

<b>all</b>	Display all spanning-tree switch debug messages.
<b>errors</b>	Display debug messages for the interface between the spanning-tree software module and the port manager software module.
<b>flush</b>	Display debug messages for the shim flush operation.
<b>general</b>	Display general event debug messages.
<b>helper</b>	Display spanning-tree helper-task debug messages. Helper tasks handle bulk spanning-tree updates.
<b>pm</b>	Display port-manager event debug messages.
<b>rx</b>	Display received bridge protocol data unit (BPDU) handling debug messages. The keywords have these meanings: <ul style="list-style-type: none"> <li>• <b>decode</b>—Display decoded received packets.</li> <li>• <b>errors</b>—Display receive error debug messages.</li> <li>• <b>interrupt</b>—Display interrupt service request (ISR) debug messages.</li> <li>• <b>process</b>—Display process receive BPDU debug messages.</li> </ul>
<b>state</b>	Display spanning-tree port state change debug messages;
<b>tx [decode]</b>	Display sent BPDU handling debug messages. The keyword has this meaning: <ul style="list-style-type: none"> <li>• <b>decode</b>—(Optional) Display decoded sent packets.</li> </ul>
<b>uplinkfast</b>	Display uplinkfast packet transmission debug messages.

### Defaults

Debugging is disabled.

### Command Modes

Privileged EXEC

### Command History

Release	Modification
12.1(11)AX	This command was introduced.
12.1(14)EA1	The <b>flush</b> and <b>uplinkfast</b> keywords were added.

**Usage Guidelines**

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

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

**Related Commands**

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
<b>show spanning-tree</b>	Displays spanning-tree state information.

# debug spanning-tree uplinkfast

Use the **debug spanning-tree uplinkfast** privileged EXEC command to enable debugging of spanning-tree UplinkFast events. Use the **no** form of this command to disable debugging output.

**debug spanning-tree uplinkfast** [exceptions]

**no debug spanning-tree uplinkfast** [exceptions]

<b>Syntax Description</b>	<b>exceptions</b> (Optional) Display spanning-tree UplinkFast-exception debug messages.
---------------------------	---

<b>Defaults</b>	Debugging is disabled.
-----------------	------------------------

<b>Command Modes</b>	Privileged EXEC
----------------------	-----------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.1(11)AX	This command was introduced.

<b>Usage Guidelines</b>	<p>The <b>undebg spanning-tree uplinkfast</b> command is the same as the <b>no debug spanning-tree uplinkfast</b> command.</p> <p>When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the <b>session</b> <i>switch-number</i> privileged EXEC command. Then enter the <b>debug</b> command at the command-line prompt of the stack member. You also can use the <b>remote command</b> <i>stack-member-number</i> <i>LINE</i> privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.</p>
-------------------------	---

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<b>show spanning-tree</b>	Displays spanning-tree state information.

# debug sw-vlan

Use the **debug sw-vlan** privileged EXEC command to enable debugging of VLAN manager activities. Use the **no** form of this command to disable debugging output.

```
debug sw-vlan {badpmcookies | cfg-vlan {bootup | cli} | events | ifs | management | notification | packets | registries | vtp}
```

```
no debug sw-vlan {badpmcookies | cfg-vlan {bootup | cli} | events | ifs | management | notification | packets | registries | vtp}
```

Syntax Description		
<b>badpmcookies</b>		Display debug messages for VLAN manager incidents of bad port manager cookies.
<b>cfg-vlan</b> {bootup   cli}		Display config-vlan debug messages. The keywords have these meanings: <ul style="list-style-type: none"> <li><b>bootup</b>—Display messages when the switch is booting up.</li> <li><b>cli</b>—Display messages when the command-line interface (CLI) is in config-vlan mode.</li> </ul>
<b>events</b>		Display debug messages for VLAN manager events.
<b>ifs</b>		See the <a href="#">debug sw-vlan ifs</a> command.
<b>management</b>		Display debug messages for VLAN manager management of internal VLANs.
<b>notification</b>		See the <a href="#">debug sw-vlan notification</a> command.
<b>packets</b>		Display debug messages for packet handling and encapsulation processes.
<b>registries</b>		Display debug messages for VLAN manager registries.
<b>vtp</b>		See the <a href="#">debug sw-vlan vtp</a> command.

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebug sw-vlan** command is the same as the **no debug sw-vlan** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<b>show vlan</b>	Displays the parameters for all configured VLANs or one VLAN (if the VLAN name or ID is specified) in the administrative domain.
	<b>show vtp</b>	Displays general information about VTP management domain, status, and counters.

# debug sw-vlan ifs

Use the **debug sw-vlan ifs** privileged EXEC command to enable debugging of the VLAN manager IOS file system (IFS) error tests. Use the **no** form of this command to disable debugging output.

```
debug sw-vlan ifs {open {read | write} | read {1 | 2 | 3 | 4} | write}
```

```
no debug sw-vlan ifs {open {read | write} | read {1 | 2 | 3 | 4} | write}
```

## Syntax Description

<b>open {read   write}</b>	Display VLAN manager IFS file-open operation debug messages. The keywords have these meanings: <ul style="list-style-type: none"> <li><b>read</b>—Display VLAN manager IFS file-read operation debug messages.</li> <li><b>write</b>—Display VLAN manager IFS file-write operation debug messages.</li> </ul>
<b>read {1   2   3   4}</b>	Display file-read operation debug messages for the specified error test (1, 2, 3, or 4).
<b>write</b>	Display file-write operation debug messages.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug sw-vlan ifs** command is the same as the **no debug sw-vlan ifs** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

When selecting the file read operation, Operation **1** reads the file header, which contains the header verification word and the file version number. Operation **2** reads the main body of the file, which contains most of the domain and VLAN information. Operation **3** reads type length version (TLV) descriptor structures. Operation **4** reads TLV data.

■ debug sw-vlan ifs

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<b>show vlan</b>	Displays the parameters for all configured VLANs or one VLAN (if the VLAN name or ID is specified) in the administrative domain.



# debug sw-vlan notification

Use the **debug sw-vlan notification** privileged EXEC command to enable debugging of the activation and deactivation of Inter-Link Switch (ISL) VLAN IDs. Use the **no** form of this command to disable debugging output.

```
debug sw-vlan notification { accfwdchange | allowedvlanfgchange | fwdchange | linkchange | modechange | pruningfgchange | statechange }
```

```
no debug sw-vlan notification { accfwdchange | allowedvlanfgchange | fwdchange | linkchange | modechange | pruningfgchange | statechange }
```

## Syntax Description

<b>accfwdchange</b>	Display debug messages for VLAN manager notification of aggregated access interface spanning-tree forward changes.
<b>allowedvlanfgchange</b>	Display debug messages for VLAN manager notification of changes to the allowed VLAN configuration.
<b>fwdchange</b>	Display debug messages for VLAN manager notification of spanning-tree forwarding changes.
<b>linkchange</b>	Display debug messages for VLAN manager notification of interface link-state changes.
<b>modechange</b>	Display debug messages for VLAN manager notification of interface mode changes.
<b>pruningfgchange</b>	Display debug messages for VLAN manager notification of changes to the pruning configuration.
<b>statechange</b>	Display debug messages for VLAN manager notification of interface state changes.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebg sw-vlan notification** command is the same as the **no debug sw-vlan notification** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
	<a href="#">show vlan</a>	Displays the parameters for all configured VLANs or one VLAN (if the VLAN name or ID is specified) in the administrative domain.

## debug sw-vlan vtp

Use the **debug sw-vlan vtp** privileged EXEC command to enable debugging of the VLAN Trunking Protocol (VTP) code. Use the **no** form of this command to disable debugging output.

```
debug sw-vlan vtp {events | packets | pruning [packets | xmit] | xmit}
```

```
no debug sw-vlan vtp {events | packets | pruning | xmit}
```

Syntax Description		
	<b>events</b>	Display debug messages for general-purpose logic flow and detailed VTP messages generated by the VTP_LOG_RUNTIME macro in the VTP code.
	<b>packets</b>	Display debug messages for the contents of all incoming VTP packets that have been passed into the VTP code from the IOS VTP platform-dependent layer, except for pruning packets.
	<b>pruning [packets   xmit]</b>	Display debug messages generated by the pruning segment of the VTP code. The keywords have these meanings: <ul style="list-style-type: none"> <li><b>packets</b>—(Optional) Display debug messages for the contents of all incoming VTP pruning packets that have been passed into the VTP code from the IOS VTP platform-dependent layer.</li> <li><b>xmit</b>—(Optional) Display debug messages for the contents of all outgoing VTP packets that the VTP code requests the IOS VTP platform-dependent layer to send.</li> </ul>
	<b>xmit</b>	Display debug messages for the contents of all outgoing VTP packets that the VTP code requests the IOS VTP platform-dependent layer to send, except for pruning packets.

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebug sw-vlan vtp** command is the same as the **no debug sw-vlan vtp** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

If no further parameters are entered after the **pruning** keyword, VTP pruning debugging messages appear. They are generated by the VTP\_PRUNING\_LOG\_NOTICE, VTP\_PRUNING\_LOG\_INFO, VTP\_PRUNING\_LOG\_DEBUG, VTP\_PRUNING\_LOG\_ALERT, and VTP\_PRUNING\_LOG\_WARNING macros in the VTP pruning code.

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
<b>show vtp</b>	Displays general information about VTP management domain, status, and counters.

# debug uddl

Use the **debug uddl** privileged EXEC command to enable debugging of the UniDirectional Link Detection (UDLD) feature. Use the **no** form of this command to disable UDLD debugging.

**debug uddl** {events | packets | registries}

**no debug uddl** {events | packets | registries}

## Syntax Description

<b>events</b>	Display debug messages for UDLD process events as they occur.
<b>packets</b>	Display debug messages for the UDLD process as it receives packets from the packet queue and tries to send them at the request of the UDLD protocol code.
<b>registries</b>	Display debug messages for the UDLD process as it processes registry calls from the UDLD process-dependent module and other feature modules.

## Defaults

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(11)AX	This command was introduced.

## Usage Guidelines

The **undebug uddl** command is the same as the **no debug uddl** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

For **debug uddl events**, these debugging messages appear:

- General UDLD program logic flow
- State machine state changes
- Program actions for the set and clear ErrDisable state
- Neighbor cache additions and deletions
- Processing of configuration commands
- Processing of link-up and link-down indications

For **debug udd packets**, these debugging messages appear:

- General packet processing program flow on receipt of an incoming packet
- Indications of the contents of the various pieces of packets received (such as type length versions [TLVs]) as they are examined by the packet reception code
- Packet transmission attempts and the outcome

For **debug udd registries**, these categories of debugging messages appear:

- Sub-block creation
- Fiber-port status changes
- State change indications from the port manager software
- MAC address registry calls

---

**Related Commands**

Command	Description
<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select
<b>show udd</b>	Displays UDD administrative and operational status for all ports or the specified port.

# debug vqpc

Use the **debug vqpc** privileged EXEC command to enable debugging of the VLAN Query Protocol (VQP) client. Use the **no** form of this command to disable debugging output.

**debug vqpc** [**all** | **cli** | **events** | **learn** | **packet**]

**no debug vqpc** [**all** | **cli** | **events** | **learn** | **packet**]

Syntax Description	all	(Optional) Display all VQP client debug messages.
	<b>cli</b>	(Optional) Display the VQP client command-line interface (CLI) debug messages.
	<b>events</b>	(Optional) Display VQP client event debug messages.
	<b>learn</b>	(Optional) Display VQP client address learning debug messages.
	<b>packet</b>	(Optional) Display VQP client packet information debug messages.

**Defaults** Debugging is disabled.

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.1(11)AX	This command was introduced.

**Usage Guidelines** The **undebg vqpc** command is the same as the **no debug vqpc** command.

When you enable debugging, it is enabled only on the stack master. To enable debugging on a stack member, you can start a session from the stack master by using the **session switch-number** privileged EXEC command. Then enter the **debug** command at the command-line prompt of the stack member. You also can use the **remote command stack-member-number LINE** privileged EXEC command on the stack master switch to enable debugging on a member switch without first starting a session.

Related Commands	Command	Description
	<b>show debugging</b>	Displays information about the types of debugging that are enabled. For syntax information, select

■ debug vqpc