# Interface and Hardware Commands

- bluetooth pin, on page 4
- clear coap database, on page 5
- clear macro auto configuration, on page 6
- coap endpoint (coap-proxy configuration), on page 7
- debug coap, on page 8
- device classifier, on page 9
- debug ilpower, on page 10
- debug interface, on page 11
- debug lldp packets, on page 12
- debug platform poe, on page 13
- debug platform software fed switch active punt packet-capture start, on page 14
- duplex, on page 15
- errdisable detect cause, on page 17
- errdisable recovery cause, on page 19
- errdisable recovery cause, on page 21
- hw-module beacon, on page 23
- interface, on page 24
- interface range, on page 26
- ip mtu, on page 28
- ipv6 mtu, on page 29
- list (coap-proxy configuration), on page 30
- lldp (interface configuration), on page 31
- logging event power-inline-status, on page 33
- macro, on page 34
- macro auto, on page 37
- macro auto apply (Cisco IOS shell scripting capability), on page 40
- macro auto config (Cisco IOS shell scripting capability), on page 42
- macro auto control, on page 43
- macro auto execute, on page 45
- macro auto global control, on page 52
- macro auto global processing, on page 54
- macro auto mac-address-group, on page 55
- macro auto processing, on page 57

- macro auto sticky, on page 58
- macro auto trigger, on page 59
- macro description, on page 60
- macro global, on page 61
- macro global description, on page 63
- max-endpoints (coap-proxy configuration), on page 64
- mdix auto, on page 65
- monitoring, on page 66
- network-policy, on page 67
- network-policy profile (global configuration), on page 68
- platform usb disable, on page 69
- port-dtls (coap-proxy configuration), on page 70
- port-unsecure (coap-proxy configuration), on page 71
- power-priority , on page 72
- power inline, on page 74
- power inline police, on page 77
- power supply, on page 79
- power supply autoLC shutdown, on page 81
- resource directory (coap-proxy configuration), on page 82
- security (coap-proxy configuration), on page 83
- shell trigger, on page 84
- show beacon all, on page 85
- show coap dtls endpoints, on page 86
- show coap endpoints, on page 87
- show coap globals, on page 88
- show coap resources, on page 89
- show coap stats, on page 90
- show coap version, on page 91
- show device classifier attached, on page 92
- show device classifier clients, on page 94
- show device classifier profile type, on page 95
- show environment, on page 98
- show errdisable detect, on page 100
- show errdisable recovery, on page 102
- show idprom tan, on page 103
- show ip interface, on page 104
- show interfaces, on page 109
- show interfaces counters, on page 115
- show interfaces switchport, on page 117
- show interfaces transceiver, on page 119
- show macro auto, on page 123
- show memory platform, on page 126
- show module, on page 129
- show network-policy profile, on page 130
- show parser macro, on page 131
- show platform hardware bluetooth, on page 134

- show platform hardware fed switch forward interface, on page 135
- show platform hardware fed switch fwd-asic counters tla, on page 138
- show platform hardware fed active fwd-asic resource tcam utilization, on page 142
- show platform resources, on page 144
- show platform software audit, on page 145
- show platform software fed switch punt cpuq rates, on page 149
- show platform software fed switch punt packet-capture display, on page 151
- show platform software fed switch punt packet-capture cpu-top-talker, on page 153
- show platform software fed switch punt rates interfaces, on page 156
- show platform software ilpower, on page 159
- show platform software memory, on page 161
- show platform software process list, on page 167
- show platform software process memory, on page 171
- show platform software process slot switch, on page 174
- show platform software status control-processor, on page 176
- show platform software thread list, on page 179
- show platform usb status, on page 181
- show processes cpu platform, on page 182
- show processes cpu platform history, on page 185
- show processes cpu platform monitor, on page 188
- show processes memory, on page 190
- show processes memory platform, on page 193
- show processes platform, on page 197
- show shell, on page 200
- show system mtu, on page 203
- show tech-support , on page 204
- show tech-support bgp, on page 206
- show tech-support diagnostic, on page 209
- speed, on page 211
- start (coap-proxy configuration), on page 213
- stop (coap-proxy configuration), on page 214
- switchport block, on page 215
- system mtu, on page 216
- transport (coap-proxy configuration), on page 217
- voice-signaling vlan (network-policy configuration), on page 218
- voice vlan (network-policy configuration), on page 220

# bluetooth pin

To configure a new Bluetooth pin, use the **bluetooth pin** command in global configuration mode.

**bluetooth pin** *pin*

## Syntax Description

| | |
|---|---|
| *pin* | Pairing pin for the Bluetooth interface.<br>The pin is a 4-digit number. |

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Gibraltar 16.12.1 | This command was introduced. |

## Usage Guidelines

The **bluetooth pin** command can be configured either in the global configuration mode. Cisco recommends using the global configuration mode to configure the Bluetooth pin.

## Examples

This example shows how to configure a new Bluetooth pin using the **bluetooth pin** command.

```
Device> enable
Device# configure terminal
Device(config)# bluetooth pin 1111
Device(config)#
```

## Related Commands

| Command | Description |
|---|---|
| **show platform hardware bluetooth** | Displays information about the Bluetooth interface |

# clear coap database

To clear the CoAP database, use the **clear coap database** command in user EXEC or privileged EXEC mode.

**clear coap database**

**Command Default**

This command has no arguments or keywords.

**Command Modes**

User EXEC (>)

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

### Example

This example shows how to clear the coap database:

```
Device(config)# clear coap database
```

# clear macro auto configuration

To remove the macro applied configuration from the interfaces, use the **clear macro auto configuration** command.

**Note** Before executing the **clear macro auto configuration** command, you must disable Auto SmartPorts on the switch.

**clear macro auto configuration**{**all** | **interface** [*interface-id*]}

**Syntax Description**

| | |
|---|---|
| *all* | Removes macro applied configuration from all the interfaces. |
| **interface** [*interface-id*] | Removes macro applied configuration from an interface. |

**Command Default** This command has no default setting.

**Command Modes** User EXEC (>)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines** Use the command to remove configuration applied by macros from all the interfaces or a particular interface on the switch.

You can verify your settings by entering the **show macro auto interface** command in privileged EXEC mode.

### Example

This example shows how to remove the configuration from all the switch interfaces:

```
Device(config)# clear macro auto configuration all
```

# coap endpoint (coap-proxy configuration)

To configure the COAP Proxy to support multiple IPv4/IPv6 static-endpoints, use the **coap endpoint** command in coap-proxy configuration mode. To return to the default settings, use the **no** form of the command.

**coap endpoint** {**ipv4** | **ipv6**}[*ip-address*]
**no coap endpoint** {**ipv4** | **ipv6**}[*ip-address*]

**Syntax Description**

| | |
|---|---|
| **ipv4** *ip-address* | Specifies IPv4 static endpoint. |
| **ipv6** *ip-address* | Specifies IPv6 static endpoint. |

**Command Modes**

coap-proxy configuration (config-coap-proxy)

**Command History**

| **Release** | **Modification** |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

### Example

This example show how to configure IPv4 static endpoint

```
Device(config)# endpoint ipv4 192.168.255.1
Device(config-coap-proxy)# transport tcp
```

# debug coap

To enable debugging of the coap configurations, use the **debug coap** command in privileged EXEC mode.

**debug coap** {**all** | **database** | **errors** | **events** | **packet** | **trace** | **warnings**}

## Syntax Description

| | |
|---|---|
| **all** | Displays all coap debug messages. |
| **database** | Displays coap database debug messages. |
| **errors** | Displays coap error debug messages. |
| **events** | Displays coap event debug messages. |
| **packet** | Displays coap packet debug messages. |
| **trace** | Displays coap trace debug messages. |
| **warnings** | Displats coap warning debug messages |

## Command Default

This command has no arguments or keywords.

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

### Example

The example shows how to enable debugging for coap database:

```
Device# debug coap database
```

# device classifier

To enable the device classifier, use the **device classifier** command in global configuration mode. Use the **no** form of this command to disable the device classifier.

**device classifier**

**no device classifier**

**Command Default**

This command is disabled by default.

**Command Modes**

Global configuration (config)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

Use the **no device classifier** command, in global configuration mode, to disable the device classifier. You cannot disable the device classifier while it is being used by features such as Auto SmartPorts (ASP).

### Example

This example shows how to enable the ASP device classifier on a switch:

```
Device(config)# device classifier
Device(config)# end
```

# debug ilpower

To enable debugging of the power controller and Power over Ethernet (PoE) system, use the **debug ilpower** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

**debug ilpower** {**cdp** | **event** | **ha** | **port** | **powerman** | **registries** | **scp** | **sense**}
**no debug ilpower** {**cdp** | **event** | **ha** | **port** | **powerman** | **registries** | **scp** | **sense**}

## Syntax Description

| | |
|---|---|
| **cdp** | Displays PoE Cisco Discovery Protocol (CDP) debug messages. |
| **event** | Displays PoE event debug messages. |
| **ha** | Displays PoE high-availability messages. |
| **port** | Displays PoE port manager debug messages. |
| **powerman** | Displays PoE power management debug messages. |
| **registries** | Displays PoE registries debug messages. |
| **scp** | Displays PoE SCP debug messages. |
| **sense** | Displays PoE sense debug messages. |

## Command Default

Debugging is disabled.

## Command Modes

Privileged EXEC

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

This command is supported only on PoE-capable switches.

When you enable debugging on a switch stack, it is enabled only on the active switch. To enable debugging on a stack member, you can start a session from the active switch by using the **session** *switch-number* 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* EXEC command on the active switc to enable debugging on a member switch without first starting a session.

# debug interface

To enable debugging of interface-related activities, use the **debug interface** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

**debug interface** {*interface-id* | **counters** {**exceptions** | **protocol memory**} | **null** *interface-number* | **port-channel** *port-channel-number* | **states** | **vlan** *vlan-id*}
**no debug interface** {*interface-id* | **counters** {**exceptions** | **protocol memory**} | **null** *interface-number* | **port-channel** *port-channel-number* | **states** | **vlan** *vlan-id*}

## Syntax Description

| | |
|---|---|
| *interface-id* | ID of the physical interface. Displays debug messages for the specified physical port, identified by type switch number/module number/port, for example, gigabitethernet 1/0/2. |
| **null** *interface-number* | Displays debug messages for null interfaces. The interface number is always **0**. |
| **port-channel** *port-channel-number* | Displays debug messages for the specified EtherChannel port-channel interface. The *port-channel-number* range is 1 to 48. |
| **vlan** *vlan-id* | Displays debug messages for the specified VLAN. The vlan range is 1 to 4094. |
| **counters** | Displays counters debugging information. |
| **exceptions** | Displays debug messages when a recoverable exceptional condition occurs during the computation of the interface packet and data rate statistics. |
| **protocol memory** | Displays debug messages for memory operations of protocol counters. |
| **states** | Displays intermediary debug messages when an interface's state transitions. |

## Command Default

Debugging is disabled.

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | 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 on a switch stack, it is enabled only on the active switch. To enable debugging on a stack member, you can start a session from the active switch by using the **session** *switch-number* 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* EXEC command on the active switch to enable debugging on a member switch without first starting a session.

# debug lldp packets

To enable debugging of Link Layer Discovery Protocol (LLDP) packets, use the **debug lldp packets** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

**debug lldp packets**
**no debug lldp packets**

**Syntax Description**

This commnd has no arguments or keywords.

**Command Default**

Debugging is disabled.

**Command Modes**

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

The **undebug lldp packets** command is the same as the **no debug lldp packets** command.

When you enable debugging on a switch stack, it is enabled only on the active switch. To enable debugging on a stack member, you can start a session from the active switch by using the **session** *switch-number* EXEC command.

# debug platform poe

To enable debugging of a Power over Ethernet (PoE) port, use the **debug platform poe** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

**debug platform poe** [{**error** | **info**}] [**switch** *switch-number*]
**no debug platform poe** [{**error** | **info**}] [**switch** *switch-number*]

**Syntax Description**

| | |
|---|---|
| **error** | (Optional) Displays PoE-related error debug messages. |
| **info** | (Optional) Displays PoE-related information debug messages. |
| **switch** *switch-number* | (Optional) Specifies the stack member. This keyword is supported only on stacking-capable switches. |

**Command Default**

Debugging is disabled.

**Command Modes**

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

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

# debug platform software fed switch active punt packet-capture start

To enable debugging of packets during high CPU utilization, for an active switch, use the **debug platform software fed switch active punt packet-capture start** command in privileged EXEC mode. To disable debugging of packets during high CPU utilization, for an active switch, use the **debug platform software fed switch active punt packet-capture stop** command in privileged EXEC mode.

**debug platform software fed switch active punt packet-capture start**
**debug platform software fed switch active punt packet-capture stop**

## Syntax Description

| | |
|---|---|
| **switch active** | Displays information about the active switch. |
| **punt** | Specifies the punt information. |
| **packet-capture** | Specifies information about the captured packet. |
| **start** | Enables debugging of the active switch. |
| **stop** | Disables debugging of the active switch. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Gibraltar 16.10.1 | This command was introduced. |

## Usage Guidelines

The **debug platform software fed switch active punt packet-capture start** command starts the debugging of packets during high CPU utilization. The packet capture is stopped when the 4k buffer size is exceeded.

## Examples

The following is a sample output from the **debug platform software fed switch active punt packet-capture start** command:

```
Device# debug platform software fed switch active packet-capture start
Punt packet capturing started.
```

The following is a sample output from the **debug platform software fed switch active punt packet-capture stop** command:

```
Device# debug platform software fed switch active packet-capture stop
Punt packet capturing stopped. Captured 101 packet(s)
```

# duplex

To specify the duplex mode of operation for a port, use the **duplex** command in interface configuration mode. To return to the default value, use the **no** form of this command.

**duplex** {**auto** | **full** | **half**}
**no duplex** {**auto** | **full** | **half**}

## Syntax Description

| | |
|---|---|
| **auto** | Enables automatic duplex configuration. The port automatically detects whether it should run in full- or half-duplex mode, depending on the attached device mode. |
| **full** | Enables full-duplex mode. |
| **half** | Enables half-duplex mode (only for interfaces operating at 10 or 100 Mb/s). You cannot configure half-duplex mode for interfaces operating at 1000 Mb/s, 10,000 Mb/s, 2.5Gb/s, or 5Gb/s. |

## Command Default

The default is **auto** for Gigabit Ethernet ports.

Duplex options are not supported on the 1000BASE-*x* or 10GBASE-*x* (where -*x* is -BX, -CWDM, -LX, -SX, or -ZX) small form-factor pluggable (SFP) modules.

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

For Gigabit Ethernet ports, setting the port to **auto** has the same effect as specifying **full** if the attached device does not autonegotiate the duplex parameter.

**Note** Half-duplex mode is supported on Gigabit Ethernet interfaces if the duplex mode is **auto** and the connected device is operating at half duplex. However, you cannot configure these interfaces to operate in half-duplex mode.

Certain ports can be configured to be either full duplex or half duplex. How this command is applied depends on the device to which the switch is attached.

If both ends of the line support autonegotiation, we highly recommend using the default autonegotiation settings. If one interface supports autonegotiation and the other end does not, configure duplex and speed on both interfaces, and use the **auto** setting on the supported side.

If the speed is set to **auto**, the switch negotiates with the device at the other end of the link for the speed setting and then forces the speed setting to the negotiated value. The duplex setting remains as configured on each end of the link, which could result in a duplex setting mismatch.

You can configure the duplex setting when the speed is set to **auto**.

⚠

**Caution** Changing the interface speed and duplex mode configuration might shut down and re-enable the interface during the reconfiguration.

You can verify your setting by entering the **show interfaces** privileged EXEC command.

## Examples

This example shows how to configure an interface for full-duplex operation:

```
Device(config)# interface gigabitethernet1/0/1
Devic(config-if)# duplex full
```

# errdisable detect cause

To enable error-disable detection for a specific cause or for all causes, use the **errdisable detect cause** command in global configuration mode. To disable the error-disable detection feature, use the **no** form of this command.

**errdisable detect cause** {**all** | **arp-inspection** | **bpduguard shutdown vlan** | **dhcp-rate-limit** | **dtp-flap** | **gbic-invalid** | **inline-power** | **link-flap** | **loopback** | **pagp-flap** | **pppoe-ia-rate-limit** | **psp shutdown vlan** | **security-violation shutdown vlan** | **sfp-config-mismatch**}
**no errdisable detect cause** {**all** | **arp-inspection** | **bpduguard shutdown vlan** | **dhcp-rate-limit** | **dtp-flap** | **gbic-invalid** | **inline-power** | **link-flap** | **loopback** | **pagp-flap** | **pppoe-ia-rate-limit** | **psp shutdown vlan** | **security-violation shutdown vlan** | **sfp-config-mismatch**}

### Syntax Description

| | |
|---|---|
| **all** | Enables error detection for all error-disabled causes. |
| **arp-inspection** | Enables error detection for dynamic Address Resolution Protocol (ARP) inspection. |
| **bpduguard shutdown vlan** | Enables per-VLAN error-disable for BPDU guard. |
| **dhcp-rate-limit** | Enables error detection for DHCP snooping. |
| **dtp-flap** | Enables error detection for the Dynamic Trunking Protocol (DTP) flapping. |
| **gbic-invalid** | Enables error detection for an invalid Gigabit Interface Converter (GBIC) module.<br>**Note** This error refers to an invalid small form-factor pluggable (SFP) module. |
| **inline-power** | Enables error detection for the Power over Ethernet (PoE) error-disabled cause.<br>**Note** This keyword is supported only on switches with PoE ports. |
| **link-flap** | Enables error detection for link-state flapping. |
| **loopback** | Enables error detection for detected loopbacks. |
| **pagp-flap** | Enables error detection for the Port Aggregation Protocol (PAgP) flap error-disabled cause. |
| **pppoe-ia-rate-limit** | Enables error detection for the PPPoE Intermediate Agent rate-limit error-disabled cause. |
| **psp shutdown vlan** | Enables error detection for protocol storm protection (PSP). |
| **security-violation shutdown vlan** | Enables voice aware 802.1x security. |
| **sfp-config-mismatch** | Enables error detection on an SFP configuration mismatch. |

**Command Default**

Detection is enabled for all causes. All causes, except per-VLAN error disabling, are configured to shut down the entire port.

**Command Modes**

Global configuration

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

A cause (such as a link-flap or dhcp-rate-limit) is the reason for the error-disabled state. When a cause is detected on an interface, the interface is placed in an error-disabled state, an operational state that is similar to a link-down state.

When a port is error-disabled, it is effectively shut down, and no traffic is sent or received on the port. For the bridge protocol data unit (BPDU) guard, voice-aware 802.1x security, and port-security features, you can configure the switch to shut down only the offending VLAN on the port when a violation occurs, instead of shutting down the entire port.

If you set a recovery mechanism for the cause by entering the **errdisable recovery** global configuration command, the interface is brought out of the error-disabled state and allowed to retry the operation when all causes have timed out. If you do not set a recovery mechanism, you must enter the **shutdown** and then the **no shutdown** commands to manually recover an interface from the error-disabled state.

For protocol storm protection, excess packets are dropped for a maximum of two virtual ports. Virtual port error disabling using the **psp** keyword is not supported for EtherChannel and Flexlink interfaces.

To verify your settings, enter the **show errdisable detect** privileged EXEC command.

This example shows how to enable error-disabled detection for the link-flap error-disabled cause:

```
Device(config)# errdisable detect cause link-flap
```

This command shows how to globally configure BPDU guard for a per-VLAN error-disabled state:

```
Device(config)# errdisable detect cause bpduguard shutdown vlan
```

This command shows how to globally configure voice-aware 802.1x security for a per-VLAN error-disabled state:

```
Device(config)# errdisable detect cause security-violation shutdown vlan
```

You can verify your setting by entering the **show errdisable detect** privileged EXEC command.

# errdisable recovery cause

To enable the error-disabled mechanism to recover from a specific cause, use the **errdisable recovery cause** command in global configuration mode. To return to the default setting, use the **no** form of this command.

**errdisable recovery cause** {**all** | **arp-inspection** | **bpduguard** | **channel-misconfig** | **dhcp-rate-limit** | **dtp-flap** | **gbic-invalid** | **inline-power** | **link-flap** | **loopback** | **mac-limit** | **pagp-flap** | **port-mode-failure** | **pppoe-ia-rate-limit** | **psecure-violation** | **psp** | **security-violation** | **sfp-config-mismatch** | **storm-control** | **udld**}
**no errdisable recovery cause** {**all** | **arp-inspection** | **bpduguard** | **channel-misconfig** | **dhcp-rate-limit** | **dtp-flap** | **gbic-invalid** | **inline-power** | **link-flap** | **loopback** | **mac-limit** | **pagp-flap** | **port-mode-failure** | **pppoe-ia-rate-limit** | **psecure-violation** | **psp** | **security-violation** | **sfp-config-mismatch** | **storm-control** | **udld**}

**Syntax Description**

| | |
|---|---|
| **all** | Enables the timer to recover from all error-disabled causes. |
| **arp-inspection** | Enables the timer to recover from the Address Resolution Protocol (ARP) inspection error-disabled state. |
| **bpduguard** | Enables the timer to recover from the bridge protocol data unit (BPDU) guard error-disabled state. |
| **channel-misconfig** | Enables the timer to recover from the EtherChannel misconfiguration error-disabled state. |
| **dhcp-rate-limit** | Enables the timer to recover from the DHCP snooping error-disabled state. |
| **dtp-flap** | Enables the timer to recover from the Dynamic Trunking Protocol (DTP) flap error-disabled state. |
| **gbic-invalid** | Enables the timer to recover from an invalid Gigabit Interface Converter (GBIC) module error-disabled state.<br>**Note** This error refers to an invalid small form-factor pluggable (SFP) error-disabled state. |
| **inline-power** | Enables the timer to recover from the Power over Ethernet (PoE) error-disabled state.<br>This keyword is supported only on switches with PoE ports. |
| **link-flap** | Enables the timer to recover from the link-flap error-disabled state. |
| **loopback** | Enables the timer to recover from a loopback error-disabled state. |
| **mac-limit** | Enables the timer to recover from the mac limit error-disabled state. |
| **pagp-flap** | Enables the timer to recover from the Port Aggregation Protocol (PAgP)-flap error-disabled state. |

| | |
|---|---|
| **port-mode-failure** | Enables the timer to recover from the port mode change failure error-disabled state. |
| **pppoe-ia-rate-limit** | Enables the timer to recover from the PPPoE IA rate limit error-disabled state. |
| **psecure-violation** | Enables the timer to recover from a port security violation disable state. |
| **psp** | Enables the timer to recover from the protocol storm protection (PSP) error-disabled state. |
| **security-violation** | Enables the timer to recover from an IEEE 802.1x-violation disabled state. |
| **sfp-config-mismatch** | Enables error detection on an SFP configuration mismatch. |
| **storm-control** | Enables the timer to recover from a storm control error. |
| **udld** | Enables the timer to recover from the UniDirectional Link Detection (UDLD) error-disabled state. |

## Command Default

Recovery is disabled for all causes.

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

A cause (such as all or BDPU guard) is defined as the reason that the error-disabled state occurred. When a cause is detected on an interface, the interface is placed in the error-disabled state, an operational state similar to link-down state.

When a port is error-disabled, it is effectively shut down, and no traffic is sent or received on the port. For the BPDU guard and port-security features, you can configure the switch to shut down only the offending VLAN on the port when a violation occurs, instead of shutting down the entire port.

If you do not enable the recovery for the cause, the interface stays in the error-disabled state until you enter the **shutdown** and the **no shutdown** interface configuration commands. If you enable the recovery for a cause, the interface is brought out of the error-disabled state and allowed to retry the operation again when all the causes have timed out.

Otherwise, you must enter the **shutdown** and then the **no shutdown** commands to manually recover an interface from the error-disabled state.

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

## Examples

This example shows how to enable the recovery timer for the BPDU guard error-disabled cause:

```
Device# Device#configure terminal
Device(config)# errdisable recovery cause bpduguard
```

# errdisable recovery cause

To enable the error-disabled mechanism to recover from a specific cause, use the **errdisable recovery cause** command in global configuration mode. To return to the default setting, use the **no** form of this command.

**errdisable recovery cause** {**all** | **arp-inspection** | **bpduguard** | **channel-misconfig** | **dhcp-rate-limit** | **dtp-flap** | **gbic-invalid** | **inline-power** | **link-flap** | **loopback** | **mac-limit** | **pagp-flap** | **port-mode-failure** | **pppoe-ia-rate-limit** | **psecure-violation** | **psp** | **security-violation** | **sfp-config-mismatch** | **storm-control** | **udld**}
**no errdisable recovery cause** {**all** | **arp-inspection** | **bpduguard** | **channel-misconfig** | **dhcp-rate-limit** | **dtp-flap** | **gbic-invalid** | **inline-power** | **link-flap** | **loopback** | **mac-limit** | **pagp-flap** | **port-mode-failure** | **pppoe-ia-rate-limit** | **psecure-violation** | **psp** | **security-violation** | **sfp-config-mismatch** | **storm-control** | **udld**}

## Syntax Description

| | |
|---|---|
| **all** | Enables the timer to recover from all error-disabled causes. |
| **arp-inspection** | Enables the timer to recover from the Address Resolution Protocol (ARP) inspection error-disabled state. |
| **bpduguard** | Enables the timer to recover from the bridge protocol data unit (BPDU) guard error-disabled state. |
| **channel-misconfig** | Enables the timer to recover from the EtherChannel misconfiguration error-disabled state. |
| **dhcp-rate-limit** | Enables the timer to recover from the DHCP snooping error-disabled state. |
| **dtp-flap** | Enables the timer to recover from the Dynamic Trunking Protocol (DTP) flap error-disabled state. |
| **gbic-invalid** | Enables the timer to recover from an invalid Gigabit Interface Converter (GBIC) module error-disabled state.<br>**Note** This error refers to an invalid small form-factor pluggable (SFP) error-disabled state. |
| **inline-power** | Enables the timer to recover from the Power over Ethernet (PoE) error-disabled state.<br>This keyword is supported only on switches with PoE ports. |
| **link-flap** | Enables the timer to recover from the link-flap error-disabled state. |
| **loopback** | Enables the timer to recover from a loopback error-disabled state. |
| **mac-limit** | Enables the timer to recover from the mac limit error-disabled state. |
| **pagp-flap** | Enables the timer to recover from the Port Aggregation Protocol (PAgP)-flap error-disabled state. |

| | |
|---|---|
| **port-mode-failure** | Enables the timer to recover from the port mode change failure error-disabled state. |
| **pppoe-ia-rate-limit** | Enables the timer to recover from the PPPoE IA rate limit error-disabled state. |
| **psecure-violation** | Enables the timer to recover from a port security violation disable state. |
| **psp** | Enables the timer to recover from the protocol storm protection (PSP) error-disabled state. |
| **security-violation** | Enables the timer to recover from an IEEE 802.1x-violation disabled state. |
| **sfp-config-mismatch** | Enables error detection on an SFP configuration mismatch. |
| **storm-control** | Enables the timer to recover from a storm control error. |
| **udld** | Enables the timer to recover from the UniDirectional Link Detection (UDLD) error-disabled state. |

## Command Default

Recovery is disabled for all causes.

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

A cause (such as all or BDPU guard) is defined as the reason that the error-disabled state occurred. When a cause is detected on an interface, the interface is placed in the error-disabled state, an operational state similar to link-down state.

When a port is error-disabled, it is effectively shut down, and no traffic is sent or received on the port. For the BPDU guard and port-security features, you can configure the switch to shut down only the offending VLAN on the port when a violation occurs, instead of shutting down the entire port.

If you do not enable the recovery for the cause, the interface stays in the error-disabled state until you enter the **shutdown** and the **no shutdown** interface configuration commands. If you enable the recovery for a cause, the interface is brought out of the error-disabled state and allowed to retry the operation again when all the causes have timed out.

Otherwise, you must enter the **shutdown** and then the **no shutdown** commands to manually recover an interface from the error-disabled state.

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

## Examples

This example shows how to enable the recovery timer for the BPDU guard error-disabled cause:

```
Device# Device#configure terminal
Device(config)# errdisable recovery cause bpduguard
```

# hw-module beacon

To control the beacon LED on a device, use the **hw-module beacon** command in the privileged EXEC mode or global configuration mode.

### Cisco IOS XE Amsterdam 17.3.x and Earlier Releases

**hw-module beacon** { **off** | **on** } **switch** *switch-number*

### Cisco IOS XE Bengaluru 17.4.1 and Later Releases

**hw-module beacon slot** { *switch-number* | **active** | **standby** } { **off** | **on** }

## Syntax Description

| | |
|---|---|
| **off** | Turns the beacon off. |
| **on** | Turns the beacon on. |
| **switch** *switch-number* | Specifies the switch to be controlled.<br>• *switch-number*: Switch number. The range is from 1 to 9. |
| **slot** {*switch-number* \| **active** \| **standby**} | Specifies the switch to be controlled.<br>• *switch-number*: Switch number. The range is from 1 to 8.<br>• **active**: Specifies the active switch.<br>• **standby**: Specifies the standby switch. |

## Command Default

This command has no default settings.

## Command Modes

Global configuration (config) (Cisco IOS XE Amsterdam 17.3.x and Earlier Releases)

Privileged EXEC (#) (Cisco IOS XE Bengaluru 17.4.1 and Later Releases)

## Command History

| **Release** | **Modification** |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |
| Cisco IOS XE Bengaluru 17.4.1 | This command was modified. |

## Usage Guidelines

Use this command to enable or disable the switch LED. Blue indicates the switch LED is on and black indicates that it is off.

The following example shows how to switch on the LED beacon of the active switch:

```
Device> enable
Device# hw-module beacon slot active on
```

# interface

To configure an interface, use the **interface** command.

**interface** {**AccessTunnel** *interface-number* | **Auto-Template** *interface-number* | **GigabitEthernet** *switch-number/slot-number/port-number* | **Internal Interface** *Internal Interface number* | **LISP** *interface-number* **Loopback** *interface-number* **Null** *interface-number* **Port-channel** *interface-number* **TenGigabitEthernet** *switch-number/slot-number/port-number* **TwentyFiveGigE** *switch-number/slot-number/port-number* **Tunnel** *interface-number* **Vlan** *interface-number* }

**Syntax Description**

| | |
|---|---|
| **AccessTunnel** *interface-number* | Enables you to configure an access tunnel interface. |
| **Auto-Template** *interface-number* | Enables you to configure a auto-template interface. The range is from 1 to 999. |
| **GigabitEthernet** *switch-number/slot-number/port-number* | Enables you to configure a Gigabit Ethernet IEEE 802.3z interface.<br>• *switch-number* — Switch ID. The range is from 1 to 8.<br>• *slot-number* — Slot number. The range is from 0 to 1.<br>• *port-number* — Port number. The range is from 1 to 48. |
| **LISP** *interface-number* | Enables you to configure a LISP interface. |
| **Loopback** *interface-number* | Enables you to configure a loopback interface. The range is from 0 to 2147483647. |
| **Null** *interface-number* | Enables you to configure a null interface. The default value is 0. |
| **Port-channel** *interface-number* | Enables you to configure a port-channel interface. The range is from 1 to 128. |
| **TenGigabitEthernet** *switch-number/slot-number/port-number* | Enables you to configure a 10-Gigabit Ethernet interface.<br>• *switch-number* — Switch ID. The range is from 1 to 8.<br>• *slot-number*<br>— Slot number. The range is from 0 to 1.<br>• *port-number* — Port number. The ranges are 1 to 4, 17 to 24, and 37 to 48. |

| | |
|---|---|
| **TwentyFiveGigE** *switch-number/slot-number/port-number* | Enables you to configure a 25-Gigabit Ethernet interface.<br>• *switch-number* — Switch ID. The range is from 1 to 8.<br>• *slot-number* — Slot number. Value is 1.<br>• *port-number* — Port number. The range is from 1 to 2. |
| **Tunnel** *interface-number* | Enables you to configure a tunnel interface. The range is from 0 to 2147483647. |
| **Vlan** *interface-number* | Enables you to configure a switch VLAN. The range is from 1 to 4094. |

## Command Default

None

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |
| Cisco IOS XE Gibraltar 16.11.1 | The **TwentyFiveGigE** keyword was added to the command. |

## Usage Guidelines

You can not use the "no" form of this command.

The range for uplink ports is 0-4.

The range for multi-Gigabit Ethernet ports on 24-port switches is 17-24.

The range for multi-Gigabit Ethernet ports on 48-port switches is 41-48.

## Examples

The following example shows how to configure a tunnel interface:

```
Device(config)# interface Tunnel 15
Device(config-if)#
```

The following example shows how to configure a 25-Gigabit Ethernet interface

```
Device(config)# interface TwentyFiveGigE 1/1/1
Device(config-if)#
```

The following example shows how to configure a 40-Gigabit Ethernet interface

# interface range

To configure an interface range, use the **interface range** command.

**interface range** { **GigabitEthernet** *switch-number/slot-number/port-number* | **Loopback** *interface-number* **Null** *interface-number* **Port-channel** *interface-number* **TenGigabitEthernet** *switch-number/slot-number/port-number* **TwentyFiveGigE** *switch-number/slot-number/port-number* **Tunnel** *interface-number* **Vlan** *interface-number* }

## Syntax Description

| | |
|---|---|
| **GigabitEthernet** *switch-number/slot-number/port-number* | Enables you to configure a Gigabit Ethernet IEEE 802.3z interface.<br>• *switch-number* — Switch ID. The range is from 1 to 8.<br>• *slot-number* — Slot number. The range is from 0 to 1.<br>• *port-number* — Port number. The range is from 0 to 48. |
| **Loopback** *interface-number* | Enables you to configure a loopback interface. The range is from 0 to 2147483647. |
| **Port-channel** *interface-number* | Enables you to configure a port-channel interface. The range is from 1 to 48. |
| **TenGigabitEthernet** *switch-number/slot-number/port-number* | Enables you to configure a 10-Gigabit Ethernet interface.<br>• *switch-number* — Switch ID. The range is from 1 to 8.<br>• *slot-number*<br>— Slot number. The range is from 0 to 1.<br>• *port-number* — Port number. The ranges are 1 to 4, 17 to 24, and 37 to 48. |
| **TwentyFiveGigE** *switch-number/slot-number/port-number* | Enables you to configure a 25-Gigabit Ethernet interface.<br>• *switch-number* — Switch ID. The range is from 1 to 8.<br>• *slot-number* — Slot number. Value is 1.<br>• *port-number* — Port number. The range is from 1 to 2. |
| **Tunnel** *interface-number* | Enables you to configure a tunnel interface. The range is from 0 to 2147483647. |

| | |
|---|---|
| **Vlan** *interface-number* | Enables you to configure a switch VLAN. The range is from 1 to 4094. |

## Command Default

None

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |
| Cisco IOS XE Gibraltar 16.11.1 | The **TwentyFiveGigE** keyword was added to the command. |

## Usage Guidelines

The range for uplink ports is 0-4.

The range for multi-Gigabit Ethernet ports on 24-port switches is 17-24.

The range for multi-Gigabit Ethernet ports on 48-port switches is 41-48.

## Examples

This example shows how you can configure interface range:

```
Device(config)# interface range vlan 1-100
```

# ip mtu

To set the IP maximum transmission unit (MTU) size of routed packets on all routed ports of the switch or switch stack, use the **ip mtu** command in interface configuration mode. To restore the default IP MTU size, use the **no** form of this command.

**ip mtu** *bytes*
**no ip mtu** *bytes*

## Syntax Description

| | |
|---|---|
| *bytes* | MTU size, in bytes. The range is from 68 up to the system MTU value (in bytes). |

## Command Default

The default IP MTU size for frames received and sent on all switch interfaces is 1500 bytes.

## Command Modes

Interface configuration (config-if)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

The upper limit of the IP value is based on the switch or switch stack configuration and refers to the currently applied system MTU value. For more information about setting the MTU sizes, see the **system mtu** global configuration command.

To return to the default IP MTU setting, you can apply the **default ip mtu** command or the **no ip mtu** command on the interface.

You can verify your setting by entering the **show ip interface** *interface-id* or **show interfaces** *interface-id* privileged EXEC command.

The following example sets the maximum IP packet size for VLAN 200 to 1000 bytes:

```
Device(config)# interface vlan 200
Device(config-if)# ip mtu 1000
```

The following example sets the maximum IP packet size for VLAN 200 to the default setting of 1500 bytes:

```
Device(config)# interface vlan 200
Device(config-if)# default ip mtu
```

This is an example of partial output from the **show ip interface** *interface-id* command. It displays the current IP MTU setting for the interface.

```
Device# show ip interface gigabitethernet4/0/1
GigabitEthernet4/0/1 is up, line protocol is up
  Internet address is 18.0.0.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set

<output truncated>
```

# ipv6 mtu

To set the IPv6 maximum transmission unit (MTU) size of routed packets on all routed ports of the switch or switch stack, use the **ipv6 mtu** command in interface configuration mode. To restore the default IPv6 MTU size, use the **no** form of this command.

**ipv6 mtu** *bytes*
**no ipv6 mtu** *bytes*

## Syntax Description

| | |
|---|---|
| *bytes* | MTU size, in bytes. The range is from 1280 up to the system MTU value (in bytes). |

## Command Default

The default IPv6 MTU size for frames received and sent on all switch interfaces is 1500 bytes.

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

The upper limit of the IPv6 MTU value is based on the switch or switch stack configuration and refers to the currently applied system MTU value. For more information about setting the MTU sizes, see the **system mtu** global configuration command.

To return to the default IPv6 MTU setting, you can apply the **default ipv6 mtu** command or the **no ipv6 mtu** command on the interface.

You can verify your setting by entering the **show ipv6 interface** *interface-id* or **show interface** *interface-id* privileged EXEC command.

The following example sets the maximum IPv6 packet size for an interface to 2000 bytes:

```
Device(config)# interface gigabitethernet4/0/1
Device(config-if)# ipv6 mtu 2000
```

The following example sets the maximum IPv6 packet size for an interface to the default setting of 1500 bytes:

```
Device(config)# interface gigabitethernet4/0/1
Device(config-if)# default ipv6 mtu
```

This is an example of partial output from the **show ipv6 interface** *interface-id* command. It displays the current IPv6 MTU setting for the interface.

```
Device# show ipv6 interface gigabitethernet4/0/1
GigabitEthernet4/0/1 is up, line protocol is up
  Internet address is 18.0.0.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set

<output truncated>
```

# list (coap-proxy configuration)

To restrict the IP address range where the lights and their resources can be learnt, use the **list** command in coap-proxy configuration mode. To return to the default settings, use the **no** form of the command.

A maximum of five ip-lists can be configured, irrespective of ipv4 or ipv6, using the **list** command.

**list** {**ipv4** | **ipv6**}[*list-name*]
**no list** {**ipv4** | **ipv6**}[*list-name*]

**Syntax Description**

| | |
|---|---|
| **ipv4** *list-name* | Specifies IPv4 list name. |
| **ipv6** *list-name* | Specifies IPv6 list name. |

**Command Modes**

coap-proxy configuration (config-coap-proxy)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

To access coap-proxy configuration mode, enter the **coap proxy** command in global configuration mode.

### Example

This example shows how to restrict the IPv4 address range using a list name.

```
Device(config)# coap proxy
Device config-coap-proxy)# list ipv4 trial_list
```

# lldp (interface configuration)

To enable Link Layer Discovery Protocol (LLDP) on an interface, use the **lldp** command in interface configuration mode. To disable LLDP on an interface, use the **no** form of this command.

**lldp** {**med-tlv-select** *tlv* | **receive** | **tlv-select** **power-management** | **transmit**}
**no lldp** {**med-tlv-select** *tlv* | **receive** | **tlv-select** **power-management** | **transmit**}

## Syntax Description

| | |
|---|---|
| **med-tlv-select** | Selects an LLDP Media Endpoint Discovery (MED) time-length-value (TLV) element to send. |
| *tlv* | String that identifies the TLV element. Valid values are the following:<br>• **inventory-management**— LLDP MED Inventory Management TLV.<br>• **location**— LLDP MED Location TLV.<br>• **network-policy**— LLDP MED Network Policy TLV.<br>• **power-management**— LLDP MED Power Management TLV. |
| **receive** | Enables the interface to receive LLDP transmissions. |
| **tlv-select** | Selects the LLDP TLVs to send. |
| **power-management** | Sends the LLDP Power Management TLV. |
| **transmit** | Enables LLDP transmission on the interface. |

## Command Default

LLDP is disabled.

## Command Modes

Interface configuration (config-if)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

This command is supported on 802.1 media types.

If the interface is configured as a tunnel port, LLDP is automatically disabled.

The following example shows how to disable LLDP transmission on an interface:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# no lldp transmit
```

The following example shows how to enable LLDP transmission on an interface:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# lldp transmit
```

# logging event power-inline-status

To enable the logging of Power over Ethernet (PoE) events, use the **logging event power-inline-status** command in interface configuration mode. To disable the logging of PoE status events, use the **no** form of this command.

**logging event power-inline-status**
**no logging event power-inline-status**

## Syntax Description

This command has no arguments or keywords.

## Command Default

Logging of PoE events is enabled.

## Command Modes

Interface configuration (config-if)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

The **no** form of this command does not disable PoE error events.

## Examples

This example shows how to enable logging of PoE events on a port:

```
Device(config-if)# interface gigabitethernet1/0/1
Device(config-if)# logging event power-inline-status
Device(config-if)#
```

# macro

To apply a macro to an interface or to apply and debug a macro on an interface, use the **macro** command in interface configuration mode.

**macro** {**apply** | **trace**}*macro-name* [**parameter** {*value*}][**parameter** {*value*}][**parameter** {*value*}]

## Syntax Description

| | |
|---|---|
| **apply** | Applies a macro to an interface. |
| **trace** | Applies a macro to an interface and then debugs it. |
| *macro-name* | Specifies the name of the macro. |
| **parameter** *value* | (Optional) Specifies unique parameter values that are specific to the interface. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive.<br>All matching occurrences of the keyword are replaced with the corresponding value. |

## Command Default

This command has no default setting.

## Command Modes

Interface configuration (config-if)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

You can use the **macro apply** *macro-name* command to apply and show the macros running on an interface.

You can use the **macro trace** *macro-name* command to apply and then debug the macro to find any syntax or configuration errors.

If a command fails because of a syntax error or a configuration error when you apply a macro, the macro continues to apply the remaining commands to the interface.

When creating a macro that requires the assignment of unique values, use the **parameter** *value* keywords to designate values specific to the interface.

Keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value. Any full match of a keyword, even if it is part of a larger string, is considered a match and is replaced by the corresponding value.

Some macros might contain keywords that require a parameter value. You can use the **macro apply** *macro-name* ? command to display a list of any required values in the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.

There are Cisco-default SmartPorts macros embedded in the switch software. You can display these macros and the commands that they contain by using the **show parser macro** command in user EXEC mode.

Follow these guidelines when you apply a Cisco-default SmartPorts macro on an interface:

- Display all macros on the switch by using the **show parser macro** command in user EXEC mode. Display the contents of a specific macro by using the **show parser macro** *macro-name* command in user EXEC mode.
- Keywords that begin with $ mean that a unique parameter value is required. Append the Cisco-default macro with the required values by using the **parameter** *value* keywords.

The Cisco-default macros use the $ character to identify required keywords. You can use the $ character to define keywords when you create a macro.

When you apply a macro to an interface, the macro name is automatically added to the interface. You can display the applied commands and macro names by using the **show running-config interface** *interface-id* command in user EXEC mode.

A macro applied to an interface range behaves the same way as a macro applied to a single interface. When you use an interface range, the macro is applied sequentially to each interface within the range. If a macro command fails on one interface, it is still applied to the remaining interfaces.

You can delete a macro-applied configuration on an interface by entering the **default interface** *interface-id* command in interface configuration mode.

### Example

After you use the **macro name** command, in interface configuration mode, you can apply it to an interface. This example shows how to apply a user-created macro called duplex to an interface:

```
Device(config-if)# macro apply duplex
```

To debug a macro, use the **macro trace** command, in interface configuration mode, to find any syntax or configuration errors in the macro as it is applied to an interface.

```
Device(config-if)# macro trace duplex
Applying command...‘duplex auto’
%Error Unknown error.
Applying command...‘speed nonegotiate’
```

This example shows how to display the Cisco-default cisco-desktop macro and how to apply the macro and set the access VLAN ID to 25 on an interface:

```
Device# show parser macro cisco-desktop
--------------------------------------------------------------
Macro name : cisco-desktop
Macro type : default
# Basic interface - Enable data VLAN only
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID
switchport mode access
# Enable port security limiting port to a single
# MAC address -- that of desktop
switchport port-security
switchport port-security maximum 1
# Ensure port-security age is greater than one minute
# and use inactivity timer
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
# Configure port as an edge network port
spanning-tree portfast
```

```
spanning-tree bpduguard enable
--------------------------------------------------------------
Device#
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# interface gigabitethernet1/0/4
Device(config-if)# macro apply cisco-desktop $AVID 25
```

# macro auto

To configure and apply a global macro using the CLI, use the **macro auto** command in privileged EXEC mode.

Use the **no** form of this command to return to the default setting.

**macro auto** {**apply** | **config**} *macro-name*

## Syntax Description

| | |
|---|---|
| **apply** | Applies the macro. |
| **config** | Enters the macro parameters. |
| *macro-name* | Specifies the macro name. |

## Command Default

No macros are applied to the switch.

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

To remove the macro from the switch, enter the **no** forms of the macro commands.

If you enter the **macro auto config** *macro-name* command, you are prompted to enter values for all the macro parameters.

Use the exact text string when entering the macro-name. The entries are case sensitive.

The user-defined values appear only in the **show macro auto** or **show running-config** command output.

### Example

This example shows how to display global macros:

```
Device# macro auto apply ?
CISCO_SWITCH_AAA_ACCOUNTING        Configure aaa accounting parameters
CISCO_SWITCH_AAA_AUTHENTICATION    Configure aaa authentication parameters
CISCO_SWITCH_AAA_AUTHORIZATION     Configure aaa authorization parameters
CISCO_SWITCH_AUTO_IP_CONFIG        Configure the ip parameters
CISCO_SWITCH_AUTO_PCI_CONFIG       Configure PCI compliant parameters
CISCO_SWITCH_DOMAIN_NAME_CONFIG    Configure domain name
CISCO_SWITCH_ETHERCHANNEL_CONFIG   Configure the etherchannel parameters
CISCO_SWITCH_HOSTNAME_CONFIG       Configure hostname
CISCO_SWITCH_HTTP_SERVER_CONFIG    Configure http server
CISCO_SWITCH_LOGGING_SERVER_CONFIG Configure logging server
CISCO_SWITCH_MGMT_VLAN_CONFIG      Configure management vlan parameters
CISCO_SWITCH_NAME_SERVER_CONFIG    Configure name server parameters
CISCO_SWITCH_NTP_SERVER_CONFIG     Configure NTP server
CISCO_SWITCH_RADIUS_SERVER_CONFIG  Configure radius server
CISCO_SWITCH_SETUP_SNMP_TRAPS      Configure SNMP trap parameters
CISCO_SWITCH_SETUP_USR_CONFIG      Configure the user parameters
CISCO_SWITCH_SNMP_SOURCE_CONFIG    Configure snmp source interface
```

```
CISCO_SWITCH_TACACS_SERVER_CONFIG   Configure tacacs server
CISCO_SWITCH_USER_PASS_CONFIG       Configure username and password
```

```
Device# macro auto config ?
CISCO_SWITCH_AAA_ACCOUNTING         Configure aaa accounting parameters
CISCO_SWITCH_AAA_AUTHENTICATION     Configure aaa authentication parameters
CISCO_SWITCH_AAA_AUTHORIZATION      Configure aaa authorization parameters
CISCO_SWITCH_AUTO_IP_CONFIG         Configure the ip parameters
CISCO_SWITCH_AUTO_PCI_CONFIG        Configure PCI compliant parameters
CISCO_SWITCH_DOMAIN_NAME_CONFIG     Configure domain name
CISCO_SWITCH_ETHERCHANNEL_CONFIG    Configure the etherchannel parameters
CISCO_SWITCH_HOSTNAME_CONFIG        Configure hostname
CISCO_SWITCH_HTTP_SERVER_CONFIG     Configure http server
CISCO_SWITCH_LOGGING_SERVER_CONFIG  Configure logging server
CISCO_SWITCH_MGMT_VLAN_CONFIG       Configure management vlan parameters
CISCO_SWITCH_NAME_SERVER_CONFIG     Configure name server parameters
CISCO_SWITCH_NTP_SERVER_CONFIG      Configure NTP server
CISCO_SWITCH_RADIUS_SERVER_CONFIG   Configure radius server
CISCO_SWITCH_SETUP_SNMP_TRAPS       Configure SNMP trap parameters
CISCO_SWITCH_SETUP_USR_CONFIG       Configure the user parameters
CISCO_SWITCH_SNMP_SOURCE_CONFIG     Configure snmp source interface
CISCO_SWITCH_TACACS_SERVER_CONFIG   Configure tacacs server
CISCO_SWITCH_USER_PASS_CONFIG       Configure username and password
```

This example shows how to display the parameters for a specific macro:

```
Device# macro auto config CISCO_SWITCH_AUTO_IP_CONFIG ?
CISCO_SWITCH_DOMAIN_NAME_CONFIG     domain name parameters
CISCO_SWITCH_LOGGING_SERVER_CONFIG  logging host parameters
CISCO_SWITCH_NAME_SERVER_CONFIG     name server parameters
CISCO_SWITCH_NTP_SERVER_CONFIG      ntp server parameters
LINE                                Provide parameters of form [Parameters
                                    name=value]
<cr>


Device# macro auto config CISCO_SWITCH_AUTO_PCI_CONFIG ?
CISCO_SWITCH_AAA_ACCOUNTING         aaa accounting parameters
CISCO_SWITCH_AAA_AUTHENTICATION     aaa authentication parameters
CISCO_SWITCH_AAA_AUTHORIZATION      aaa authorization parameters
CISCO_SWITCH_HTTP_SERVER_CONFIG     http server parameters
CISCO_SWITCH_RADIUS_SERVER_CONFIG   radius server parameters
CISCO_SWITCH_TACACS_SERVER_CONFIG   tacacs server parameters
LINE                                Provide parameters of form [Parameters
                                    name=value]
<cr>


Device# macro auto config CISCO_SWITCH_SETUP_SNMP_TRAPS ?
CISCO_SWITCH_SNMP_SOURCE_CONFIG     snmp source parameters
LINE                                Provide parameters of form [Parameters
                                    name=value]
<cr>


Device# macro auto config CISCO_SWITCH_SETUP_USR_CONFIG ?CISCO_AUTO_TIMEZONE_CONFIG timezone
 parameters
CISCO_SWITCH_HOSTNAME_CONFIG        hostname parameter
LINE                                Provide parameters of form [Parameters
                                    name=value]
<cr>
```

This example shows how to set macro parameters and apply the macro using the CLI:

```
Device# macro auto config CISCO_SWITCH_ETHERCHANNEL_CONFIG
Enter the port channel id[1-48] for 3K & 2350,[1-6] for 2K: 2
Enter the port channel type, Layer:[2-3(L3 not supported on 2K)]: 2
Enter etherchannel mode for the interface[auto/desirable/on/active/passive]: active
Enter the channel protocol[lacp/none]: lacp
Enter the number of interfaces to join the etherchannel[8-PAGP/MODE:ON,16-LACP]: 7
Enter interface name[GigabitEthernet3/0/3]: gigabitethernet1/0/1
Enter interface name[GigabitEthernet3/0/3]: gigabitethernet1/0/2
Enter interface name[GigabitEthernet3/0/3]: gigabitethernet1/0/3
Enter interface name[GigabitEthernet3/0/3]: gigabitethernet1/0/4
Enter interface name[GigabitEthernet3/0/3]: gigabitethernet1/0/5
Enter interface name[GigabitEthernet3/0/3]: gigabitethernet1/0/6
Enter interface name[GigabitEthernet3/0/3]: gigabitethernet1/0/7
Do you want to apply the parameters? [yes/no]: yes
Enter configuration commands, one per line. End with CNTL/Z.
Enter configuration commands, one per line. End with CNTL/Z.
Enter configuration commands, one per line. End with CNTL/Z.
Enter configuration commands, one per line. End with CNTL/Z.
Enter configuration commands, one per line. End with CNTL/Z.
Enter configuration commands, one per line. End with CNTL/Z.
Enter configuration commands, one per line. End with CNTL/Z.
Device# macro auto apply CISCO_SWITCH_ETHERCHANNEL_CONFIG
Enter configuration commands, one per line. End with CNTL/Z.
Device#
```

# macro auto apply (Cisco IOS shell scripting capability)

To configure and apply a global macro using the Cisco IOS shell scripting capability, use the **macro auto apply** command in privileged EXEC mode. Use the **no** form of this command to return to the default setting.

**macro auto apply** *macro-name*

**Syntax Description**

| | |
|---|---|
| **apply** | Applies the macro. |
| *macro-name* | Specifies the macro name. |

**Command Default**

No macros are applied to the switch.

**Command Modes**

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

To remove the macro from the switch, enter the **no** forms of the macro commands.

Use the exact text string when entering the *macro-name*. The entries are case sensitive.

The user-defined values appear only in the **show macro auto** or **show running-config** command output.

You can also use the Cisco IOS shell scripting capability to set the parameters. For examples, see the

“Configuring and Applying Global Macros” section in the “Configuring Auto Smartports and Static Smartports Macros” chapter.

### Example

This example shows how to display global macros:

```
Device# macro auto apply ?

CISCO_SWITCH_AAA_ACCOUNTING        Configure aaa accounting parameters
CISCO_SWITCH_AAA_AUTHENTICATION    Configure aaa authentication parameters
CISCO_SWITCH_AAA_AUTHORIZATION     Configure aaa authorization parameters
CISCO_SWITCH_AUTO_IP_CONFIG        Configure the ip parameters
CISCO_SWITCH_AUTO_PCI_CONFIG       Configure PCI compliant parameters
CISCO_SWITCH_DOMAIN_NAME_CONFIG    Configure domain name
CISCO_SWITCH_ETHERCHANNEL_CONFIG   Configure the etherchannel parameters
CISCO_SWITCH_HOSTNAME_CONFIG       Configure hostname
CISCO_SWITCH_HTTP_SERVER_CONFIG    Configure http server
CISCO_SWITCH_LOGGING_SERVER_CONFIG Configure logging server
CISCO_SWITCH_MGMT_VLAN_CONFIG      Configure management vlan parameters
CISCO_SWITCH_NAME_SERVER_CONFIG    Configure name server parameters
CISCO_SWITCH_NTP_SERVER_CONFIG     Configure NTP server
CISCO_SWITCH_RADIUS_SERVER_CONFIG  Configure radius server
CISCO_SWITCH_SETUP_SNMP_TRAPS      Configure SNMP trap parameters
CISCO_SWITCH_SETUP_USR_CONFIG      Configure the user parameters
CISCO_SWITCH_SNMP_SOURCE_CONFIG    Configure snmp source interface
```

```
CISCO_SWITCH_TACACS_SERVER_CONFIG   Configure tacacs server
CISCO_SWITCH_USER_PASS_CONFIG       Configure username and password
```

# macro auto config (Cisco IOS shell scripting capability)

To configure and apply a global macro, use the **macro auto config** command in privileged EXEC mode. Use the **no** form of this command to return to the default setting.

**macro auto config** *macro-name* [*parameter=value* [*parameter=value*]**...**]

**Syntax Description**

| | |
|---|---|
| **config** | Enters the macro parameters. |
| *macro-name* | Specifies the macro name. |
| *parameter=value [parameter=value]* ... | *parameter=value*—Replaces values for global macro parameter values. Enter values in the form of name value pair separated by a space: <name1>=<value1> [<name2>=<value2>...] |

**Command Default**

No macros are applied to the switch.

**Command Modes**

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

To remove the macro from the switch, enter the **no** forms of the macro commands.

If you enter the **macro auto config** *macro-name* command, you are prompted to enter values for all the macro parameters.

Use the exact text string when entering the *macro-name* and *parameters*. The entries are case sensitive.

The user-defined values appear only in the **show macro auto** or **show running-config** command output.

You can also use the Cisco IOS shell scripting capability to set the parameters. For examples, see the “Configuring and Applying Global Macros” section in the “Configuring Auto Smartports and Static Smartports Macros” chapter.

# macro auto control

To specify when the switch applies an Auto Smartports macro based on the detection method, device type, or trigger (referred to as event trigger control), use the **macro auto control** command in interface configuration mode. Use the **no** form of this command to disable trigger-to-macro mapping. The switch then does not apply macros based on event triggers.

**macro auto control** {**detection** [**cdp**] [**lldp**] [**mac-address**] | **device** [**ip-camera**] [**media-player**] [**phone**] [**lightweight-ap**] [**access-point**] [**router**] [**switch**] | **trigger** [**last-resort**]}
**no macro auto control** {**detection** [**cdp**] [**lldp**] [**mac-address**] | **device** [**ip-camera**] [**media-player**] [**phone**] [**lightweight-ap**] [**access-point**] [**router**] [**switch**] | **trigger** [**last-resort**]}

## Syntax Description

| | |
|---|---|
| **detection [cdp] [lldp] [mac-address]** | detection—Sets one or more of these as an event trigger:<br>• (Optional) **cdp**—CDP messages<br>• (Optional) **lldp**—LLDP messages<br>• (Optional) **mac-address**—User-defined MAC address groups |
| **device [access-point] [ip-camera] [lightweight-ap] [media-player] [phone] [router] [switch]** | device—Sets one or more of these devices as an event trigger:<br>• (Optional) **access-point**—Autonomous access point<br>• (Optional) **ip-camera**—Cisco IP video surveillance camera<br>• (Optional) **lightweight-ap**—Lightweight access point<br>• (Optional) **media-player**—Digital media player<br>• (Optional) **phone**—Cisco IP phone<br>• (Optional) **router**—Cisco router<br>• (Optional) **switch**—Cisco switch |
| **trigger [last-resort]** | trigger—Sets a specific event trigger.<br>• (Optional) **last-resort**—Last-resort trigger. |

**Command Default**

The switch uses the device type as the event trigger. If the switch cannot determine the device type, it uses MAC address groups, MAB messages, 802.1x authentication messages, and LLDP messages in random order.

**Command Modes**

Interface configuration (config-if)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

If you do not set event triggers, the switch uses the device type as the event trigger. If the switch cannot determine the device type, it uses MAC address groups, MAB messages, 802.1x authentication messages, and LLDP messages in random order.

To verify that a macro is applied to an interface, use the **show macro auto interface** command in user EXEC mode.

### Example

This example shows how to set LLDP messages and MAC address groups as event triggers:

```
Device(config)# interface gigabitethernet 5/0/2
Device(config-if)# macro auto control detection lldp mac-address
Device(config-if)# exit
Device(config)# end
```

This example shows how to set access points, video surveillance cameras, and digital media players as event triggers:

**Note** The switch applies a built-in macro only when it detects an access point, video surveillance camera, or digital media player.

```
Device(config)# interface gigabitethernet 5/0/1
Device(config-if)# macro auto control device access-point ip-camera media-player
Device(config-if)# exit
Device(config)# end
```

# macro auto execute

To replace built-in macro default values and to configure mapping from an event trigger to a built-in or user-defined macro, use the **macro auto execute** command in global configuration mode.

**macro auto execute** *event trigger* {**builtin** *built-in macro* | **remote** *url*} {*parameter=value*} {*function contents*}
**no macro auto execute** *event trigger* {**builtin** *built-in macro* | **remote** *url*} {*parameter=value*} {*function contents*}

**Syntax Description**

| | |
|---|---|
| *event trigger* | Defines mapping from an event trigger to a built-in macro.<br>Specifies an event trigger:<br>• CISCO_CUSTOM_EVENT<br>• CISCO_DMP_EVENT<br>• CISCO_IPVSC_EVENT<br>• CISCO_LAST_RESORT_EVENT<br>• CISCO_PHONE_EVENT<br>• CISCO_ROUTER_EVENT<br>• CISCO_SWITCH_EVENT<br>• CISCO_WIRELESS_AP_EVENT<br>• CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT<br>• WORD—Apply a user-defined event trigger such as a MAC address group |

| | |
|---|---|
| **builtin** *built-in macro name* | (Optional) Specifies a builtin built-in macro name:<br>• CISCO_AP_AUTO_SMARTPORT<br>Specify the parameter value: NATIVE_VLAN=1<br>• CISCO_DMP_AUTO_SMARTPORT<br>Specify the parameter value: ACCESS_VLAN=1.<br>• CISCO_IPVSC_AUTO_SMARTPORT<br>Specify the parameter value: ACCESS_VLAN=1.<br>• CISCO_LWAP_AUTO_SMARTPORT<br>Specify the parameter value: ACCESS_VLAN=1.<br>• CISCO_PHONE_AUTO_SMARTPORT<br>Specify the parameter values: ACCESS_VLAN=1 and VOICE_VLAN=2.<br>• CISCO_ROUTER_AUTO_SMARTPORT<br>Specify the parameter value: NATIVE_VLAN=1.<br>• CISCO_SWITCH_AUTO_SMARTPORT<br>Specify the parameter value: NATIVE_VLAN=1. |
| *parameter=value* | (Optional) *parameter=value*—Replaces default values for parameter values shown for the *bultin-macro name*, for example, ACCESS_VLAN=1. Enter new values in the form of name value pair separated by a space: [<name1>=<value1> <name2>=<value2>...]. |
| *{function contents}* | (Optional) *{function contents}*— Specifies a user-defined macro to associate with the trigger. Enter the macro contents within braces. Begin the Cisco IOS shell commands with the left brace and end the command grouping with the right brace. |

| | |
|---|---|
| **remote** *url* | (Optional) Specifies a remote server location:<br>- The syntax for the local flash file system on the standalone switch or the stack's active switch: **flash**:<br>The syntax for the local flash file system on a stack member:<br>**flash** *member number*:<br>The syntax for the FTP:<br>**ftp**:[[//*username*[:*password*]@*location*]/*directory*]/*filename*<br>The syntax for an HTTP server:<br>**http**://[[*username*:*password*]@]{*hostname* \| *host-ip*}[/*directory*]/*filename*<br>The syntax for a secure HTTP server:<br>**https**://[[*username*:*password*]@]{*hostname* \| *host-ip*}[/*directory*]/*filename*<br>The syntax for the NVRAM:<br>**nvram**://[[*username*:*password*]@][/*directory*]/*filename*<br>The syntax for the Remote Copy Protocol (RCP):<br>**rcp**:[[//*username*@*location*]/*directory*]/*filename*<br>The syntax for the Secure Copy Protocol (SCP):<br>**scp**:[[//*username*@*location*]/*directory*]/*filename*<br>The syntax for the TFTP:<br>**tftp**:[[//*location*]/*directory*]/*filename* |

## Command Default

None

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use the **macro auto execute** command to replace the built-in macro default values with values that are specific to your switch.

The switch automatically maps from event triggers to built-in macros. The built-in macros are system-defined macros in the software image. You can also create user-defined macros by using the Cisco IOS shell scripting capability.

You can create new event triggers by using the **shell trigger** commands in global configuration mode. Use the **show shell triggers** command in privileged EXEC to display the contents of the user-defined triggers and macros.

You can use the **macro auto mac-address-group** command in global configuration mode to create event triggers for devices that do not support Cisco Discovery Protocol (CDP) or Link Layer Discovery Protocol (LLDP).

You can use the remote macro feature to store macros in a central location for designated network switches to use. You can then maintain and update the macro files for use by multiple switches. Use **remote** *url* to configure the remote server location and macro path information. There are no specific file extension requirements for saved macro files.

Auto Smartports macros and antimacros (the antimacro is the portion of the applied macro that removes it at link down) have these guidelines and limitations:

- You can delete or change the built-in macros. However, you can override a built-in macro by creating a user-defined macro with the same name. To restore the original built-in macro, delete the user-defined macro.
- If you enable both the **macro auto device** and the **macro auto execute** commands, the parameters specified in the command last executed are applied to the switch. Only one command is active on the switch.
- To avoid system conflicts when macros are applied, remove all port configurations except for 802.1x authentication.
- Do not configure port security when enabling Auto SmartPorts on the switch.
- If the macro conflicts with the original configuration, either the macro does not apply some of the original configuration commands, or the antimacro does not remove them. (The antimacro is the portion of the applied macro that removes the macro at a link-down event.)
- For example, if 802.1x authentication is enabled, you cannot remove the switchport-mode access configuration. Remove the 802.1x authentication before removing the switchport mode configuration.
- A port cannot be a member of an EtherChannel when you apply Auto SmartPorts macros.
- The built-in-macro default data VLAN is VLAN 1. The default voice VLAN is VLAN 2. If your switch uses different access, native, or voice VLANs, use the **macro auto device** or the **macro auto execute** commands to configure the values.
- For 802.1x authentication or MAC authentication bypass (MAB), to detect non-Cisco devices, configure the RADIUS server to support the Cisco attribute-value pair **auto-smart-port=***event trigger*
- The switch supports Auto SmartPort macros only on directly connected devices. Multiple device connections, such as hubs, are not supported.
- If authentication is enabled on a port, the switch ignores a MAC address trigger if authentication fails.
- The order of CLI commands within the macro and the corresponding antimacro can be different.

.

### Example

This example shows how to use two built-in macros for connecting Cisco switches and Cisco IP phones to the switch. This example modifies the default voice VLAN, access VLAN, and native VLAN for the trunk interface:

```
Device(config)# !!! the next command modifies the access and voice vlans
Device(config)# !!! for the built in Cisco IP phone auto smartport macro
Device(config)# macro auto execute CISCO_PHONE_EVENT builtin CISCO_PHONE_AUTO_SMARTPORT
ACCESS_VLAN=10 VOICE_VLAN=20
Device(config)# !!! the next command modifies the Native vlan used for inter switch trunks
Device(config)# macro auto execute CISCO_SWITCH_EVENT builtin CISCO_SWITCH_AUTO_SMARTPORT
NATIVE_VLAN=10
Device(config)# !!! the next command enables auto smart ports globally
Device(config)# macro auto global processing
Device(config)# exit
Device# !!! here is the running configuration of the interface connected
Device# !!! to another Cisco Switch after the Macro is applied
Device# show running-config interface gigabitethernet1/0/1
Building configuration...

Current configuration : 284 bytes
!
interface GigabitEthernet1/0/1
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 10
 switchport mode trunk
 srr-queue bandwidth share 10 10 60 20
 queue-set 2
 priority-queue out
 mls qos trust cos
 auto qos voip trust
 macro description CISCO_SWITCH_EVENT
end
```

This example shows how to map a user-defined event trigger called media player to a user-defined macro

1. Connect the media player to an 802.1x- or MAB-enabled switch port.
2. On the RADIUS server, set the attribute-value pair to auto-smart-port=DMP_EVENT
3. On the switch, create the event trigger DMP_EVENT, and enter the user-defined macro commands.
4. The switch recognizes the attribute-value pair=DMP_EVENT response from the RADIUS server and applies the macro associated with this event trigger.

```
Device(config)# shell trigger DMP_EVENT mediaplayer
Device(config)# macro auto execute DMP_EVENT {
if [[ $LINKUP == YES ]]; then
conf t
 interface $INTERFACE
   macro description $TRIGGER
   switchport access vlan 1
   switchport mode access
   switchport port-security
   switchport port-security maximum 1
   switchport port-security violation restrict
   switchport port-security aging time 2
   switchport port-security aging type inactivity
   spanning-tree portfast
   spanning-tree bpduguard enable
   exit
fi
if [[ $LINKUP == NO ]]; then
conf t
interface $INTERFACE
     no macro description $TRIGGER
     no switchport access vlan 1
     if [[ $AUTH_ENABLED == NO ]]; then
        no switchport mode access
     fi
     no switchport port-security
     no switchport port-security maximum 1
     no switchport port-security violation restrict
     no switchport port-security aging time 2
     no switchport port-security aging type inactivity
     no spanning-tree portfast
     no spanning-tree bpduguard enable
     exit
fi
```

*Table 1: Supported Cisco IOS Shell Keywords*

| **Command** | **Description** |
|---|---|
| { | Begin the command grouping. |
| } | End the command grouping. |
| [[ | Use as a conditional construct. |
| ]] | Use as a conditional construct. |
| else | Use as a conditional construct. |
| == | Use as a conditional construct. |
| fi | Use as a conditional construct. |
| if | Use as a conditional construct. |
| then | Use as a conditional construct. |
| -z | Use as a conditional construct. |
| $ | Variables that begin with the $ character are replaced with a parameter value. |
| # | Use the # character to enter comment text. |

*Table 2: Unsupported Cisco IOS Shell Reserved Keywords*

| **Command** | **Description** |
|---|---|
| \| | Pipeline. |
| case | Conditional construct. |
| esac | Conditional construct. |
| for | Looping construct. |
| function | Shell function. |
| in | Conditional construct. |
| select | Conditional construct. |

| Command | Description |
|---|---|
| time | Pipeline. |
| until | Looping construct. |
| while | Looping construct. |

# macro auto global control

To specify when the switch applies an Auto Smartports macro based on the device type or trigger (referred to as event trigger control), use the **macro auto global control** command in global configuration mode. Use the **no** form of this command to disable trigger-to-macro mapping.

**macro auto global control** {**detection** [**cdp**] [**lldp**][**mac-address**] | **device** [**access-point**] [**ip-camera**] [**lightweight-ap**] [**media-player**] [**phone**] [**router**] [**switch**] | **trigger** [**last-resort**]}
**no macro auto global control** {**detection** [**cdp**] [**lldp**] [**mac-address**] | **device** [**access-point**] [**ip-camera**] [**lightweight-ap**] [**media-player**] [**phone**] [**router**] [**switch**] | **trigger** [**last-resort**]}

## Syntax Description

| | |
|---|---|
| **detection [cdp] [lldp] [mac-address]** | detection—Sets one or more of these as an event trigger:<br>• (Optional) **cdp**—CDP messages<br>• (Optional) **lldp**—LLDP messages<br>• (Optional) **mac-address**—User-defined MAC address groups |
| **device [access-point] [ip-camera] [lightweight-ap] [media-player] [phone] [router] [switch]** | device—Sets one or more of these devices as an event trigger:<br>• (Optional) **access-point**—Autonomous access point<br>• (Optional) **ip-camera**—Cisco IP video surveillance camera<br>• (Optional) **lightweight-ap**—Lightweight access point<br>• (Optional) **media-player**—Digital media player<br>• (Optional) **phone**—Cisco IP phone<br>• (Optional) **router**—Cisco router<br>• (Optional) **switch**—Cisco switch |
| **trigger [last-resort]** | trigger—Sets a specific event trigger.<br>• (Optional) **last-resort**—Last-resort trigger. |

## Command Default

The switch uses the device type as the event trigger. If the switch cannot determine the device type, it uses MAC address groups, MAB messages, 802.1x authentication messages, and LLDP messages in random order.

**Command Modes**

Global configuration (config)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

If you do not set event triggers, the switch uses the device type as the event trigger. If the switch cannot determine the device type, it uses MAC address groups, MAB messages, 802.1x authentication messages, and LLDP messages in random order.

To verify that a macro is applied to a switch, use the **show macro auto global** command in user EXEC mode.

### Example

This example shows how to set CDP messages, LLDP messages and MAC address groups as event triggers:

```
Device(config)# macro auto global control detection cdp lldp mac-address
Device(config)# end
```

This example shows how to set autonomous access points, lightweight access points, and IP phones:

```
Device(config)# macro auto global control device access-point lightweight-ap phone
Device(config)# end
```

# macro auto global processing

To enable Auto SmartPorts macros on the switch, use the **macro auto global processing** command in global configuration mode. Use the **no** form of this command to disable the macros.

**macro auto global processing**

**no macro auto global processing**

**Command Default**

Auto Smartports is disabled.

**Command Modes**

Global configuration (config)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

Use the **macro auto global processing** command to globally enable macros on the switch. To disable macros on a specific port, use the **no macro auto processing** command in interface mode.

When using 802.1x or MAB authentication, you need to configure the RADIUS server to support the Cisco attribute-value pair **auto-smart-port**=*event trigger*. If authentication fails, the macro is not applied. If the 802.1x or MAB authentication fails on the interface, the switch does not use the fallback CDP event trigger.

When CDP-identified devices advertise multiple capabilities, the switch chooses a capability first by switch and then by router.

To verify that a macro is applied to an interface, use the **show macro auto interface**command in privileged EXEC mode.

### Example

This example shows how to enable Auto SmartPorts on the switch and to disable the feature on a specific interface:

```
Device(config)# macro auto global processing
Device(config)# interface gigabitethernet 0/1
Device(config-if)# no macro auto processing
Device(config-if)# exit
Device(config)#
```

# macro auto mac-address-group

To create an event trigger for devices that do not support Cisco Discovery Protocol (CDP) or Link Layer Discover Protocol (LLDP), use the **macro auto mac-address-group** command in global configuration mode. Use the **no** form of this command to delete the group.

**macro auto mac-address-group** *name* {**mac-address list** *list* | **oui** {**list** *list* | **range** *start-value* **size** *number*}}
**no macro auto mac-address-group** *name* {**mac-address list** *list* | **oui** {**list** *list* | **range** *start-value* **size** *number*}}

## Syntax Description

| | |
|---|---|
| *name* | Specifies the group name. |
| **ui** | (Optional) Specifies an operationally unique identifier (OUI) **list** or **range**.<br>• **list**—Enter an OUI list in hexadecimal format separated by spaces.<br>• **range**—Enter the starting OUI hexadecimal value (*start-value*).<br>• **size**—Enter the length of the range (number) from 1 to 5 to create a list of sequential addresses. |
| **mac-address list** *list* | (Optional) Configures a list of MAC addresses separated by a space. |

## Command Default

No groups are defined.

## Command Modes

Group configuration (config-addr-grp-mac)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use the **macro auto mac-address-group** command to create an event trigger for devices that do not support CDP or LLDP. Use the MAC address group as a trigger to map to a built-in or user-defined macro by using the **macro auto execute** command. At link-up the switch detects the device type and applies the specified macro.

The switch supports up to ten MAC address groups. Each group can have up to 32 OUI and 32 MAC configured addresses.

### Example

This example shows how to create a MAC-address-group event trigger called *address_trigger* and how to verify your entries:

```
Device(config)# macro auto mac-address-group mac address_trigger
Device(config-addr-grp-mac)# mac-address list 2222.3333.3334 22.33.44 a.b.c
Device(config-addr-grp-mac)# oui list 455555 233244
Device(config-addr-grp-mac)# oui range 333333 size 2
Device(config-addr-grp-mac)# exit
Device(config)# end
Device# show running configuration
!
!macro auto mac-address-group address_trigger
 oui list 333334
 oui list 333333
 oui list 233244
 oui list 455555
 mac-address list 000A.000B.000C
 mac-address list 0022.0033.0044
 mac-address list 2222.3333.3334
!

<output truncated>
```

# macro auto processing

To enable Auto SmartPorts macros on an interface, use the **macro auto processing** command in interface configuration mode. Use the no form of this command to disable the macros.

**macro auto processing**

**no macro auto processing**

## Command Default

Auto SmartPorts is disabled.

## Command Modes

Interface configuration (config-if)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use the **macro auto processing** command, in interface configuration mode, to enable macros on a specific interface. To disable macros on a specific interface, use the no macro auto processing command, in interface configuration mode.

A port cannot be a member of an EtherChannel when you apply Auto SmartPorts macros. If you use EtherChannels, disable Auto SmartPorts on the EtherChannel interface by using the **no macro auto processing** command. The EtherChannel interface applies the configuration to the member interfaces.

To verify that a macro is applied to an interface, use the **show macro auto interface** command in privileged EXEC mode.

### Example

This example shows how to enable Auto SmartPorts on the switch and to disable the feature on a specific interface:

```
Device(config)# interface gigabitethernet 0/1
Device(config-if)# no macro auto processing
Device(config-if)# exit
Device(config)# macro auto global processing
```

# macro auto sticky

To configure macros to remain active after a link-down event, referred to as macro persistence, use the **macro auto sticky** command in global configuration mode. Use the **no** form of this command to disable the macro persistence.

**macro auto sticky**
**no macro auto sticky**

## Command Default

Macro persistence is disabled.

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use the **macro auto sticky** command so that macros remain active after a link-down event.

### Example

This example shows how to enable macro persistence on an interface:

```
Device(config)# interface gigabitethernet 5/0/2
Device(config-if)# macro auto port sticky
Device(config-if)# exit
Device(config)# end
```

# macro auto trigger

To enter the configure-macro-trigger mode and define a trigger for a device that has no built-in trigger and associate the trigger with a device or profile, use the **macro auto trigger** command in global configuration mode. To remove the user-defined trigger, use the **no** form of this command.

**macro auto trigger** *trigger_name* {**device** | **exit** | **no** | **profile**}
**no macro auto trigger** *trigger_name* {**device** | **exit** | **no** | **profile**}

## Syntax Description

| | |
|---|---|
| *trigger_name* | Specifies a trigger to be associated with the device type or profile name. |
| **device** | Specifies a device name to map to the named trigger. |
| **exit** | Exits device group configuration mode. |
| **no** | Removes any configured device. |
| **profile** | Specifies a profile name to map to the named trigger. |

## Command Default

No user-defined triggers are configured.

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

If a device is classified by the Device Classifier, but does not have a built-in trigger defined, use the **macro auto trigger** command, in global configuration mode, to define a trigger based on a device name or a profile name. After you enter the command, the switch is in the configure-macro-trigger mode and the **device**, **exit**, **no**, and **profile** keywords are visible. In this mode, you can provide a device name or a profile name to map to the trigger. It is not necessary to map the trigger to both a device name and a profile name. If you map the trigger to both names, the trigger-to-profile name mapping has preference for macro application.

You must use this command to configure a trigger when you configure a user-defined macro. The trigger name is required for the custom macro configuration.

After the device is profiled, you must add the complete string to the device-group database.

### Example

This example shows how to configure a user-defined trigger for a profile called DMP_EVENT mediaplayer for use with a media player that has no built-in trigger:

```
Device(config)# macro auto trigger DMP
Device(config-macro-trigger)# profile mediaplayer-DMP
Device(config-macro-trigger)# exit
```

# macro description

To enter a description about which macros are applied to an interface, use the **macro description** command in interface configuration mode. Use the **no** form of this command to remove the description. This command is mandatory for Auto SmartPorts to work.

**macro description** *text*
**no macro description** *text*

**Syntax Description**

| | |
|---|---|
| **description** *text* | Enters a description about the macros that are applied to the specified interface. |

**Command Default**

This command has no default setting.

**Command Modes**

Interface configuration (config-if)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

Use the **description** keyword to associate comment text or the macro name with an interface. When multiple macros are applied on a single interface, the description text is from the last applied macro.

You can verify your settings by entering the **show parser macro description** command in privileged EXEC mode.

### Example

This example shows how to add a description to an interface:

```
(config-if)# macro description duplex settings
```

# macro global

To apply a macro to a switch or to apply and debug a macro on a switch, use the **macro global** command in global configuration mode.

**macro global** {**apply** | **trace**} *macro-name* [**parameter** {*value*}][**parameter** {*value*}][**parameter** {*value*}] parameter

## Syntax Description

| | |
|---|---|
| **apply** | Applies a macro to the switch. |
| **trace** | Applies a macro to a switch and debugs the macro. |
| *macro-name* | Specifies the name of the macro. |
| **parameter** *value* | (Optional) Specifies unique parameter values that are specific to the switch. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value. |

## Command Default

This command has no default setting.

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

**Note** You can delete a global macro-applied configuration on a switch only by entering the no version of each command in the macro.

Use the **macro global apply** *macro-name* command to apply the macro to an interface.

Use the **macro global trace** *macro-name* command to apply and then debug the macro to find any syntax or configuration errors.

If a command fails when you apply a macro because of a syntax error or a configuration error, the macro continues to apply the remaining commands to the switch.

When creating a macro that requires the assignment of unique values, use the **parameter** *value* keywords to designate values specific to the switch.

Keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value. Any full match of a keyword, even if it is part of a larger string, is considered a match and is replaced by the corresponding value.

Some macros might contain keywords that require a parameter value. You can use the **macro global apply** *macro-name* ? command to display a list of any required values in the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.

There are Cisco-default Smartports macros embedded in the switch software. You can display these macros and the commands they contain by using the **show parser macro** command in user EXEC mode.

Follow these guidelines when you apply a Cisco-default Smartports macro on a switch:

- Display all macros on the switch by using the **show parser macro** command. Display the contents of a specific macro by using the **show parser macro name** *macro-name* command.
- Keywords that begin with $ mean that a unique parameter value is required. Append the Cisco-default macro with the required values by using the **parameter** *value* keywords.

  The Cisco-default macros use the $ character to help identify required keywords. There is no restriction on using the $ character to define keywords when you create a macro.

When you apply a macro to a switch, the macro name is automatically added to the switch. You can display the applied commands and macro names by using the **show running-config** command.

### Example

After you have created a new macro by using the **macro auto execute** command, you can apply it to a switch. This example shows how to view the **snmp** macro, how to apply the macro, set the hostname to test-server, and set the IP precedence value to 7:

```
Device# show parser macro name snmp
Macro name : snmp
Macro type : customizable

#enable port security, linkup, and linkdown traps
snmp-server enable traps port-security
snmp-server enable traps linkup
snmp-server enable traps linkdown
#set snmp-server host
snmp-server host ADDRESS
#set SNMP trap notifications precedence
snmp-server ip precedence VALUE

--------------------------------------------------
Device(config)# macro global apply snmp ADDRESS test-server VALUE 7
```

To debug a macro, use the **macro global trace** command to find any syntax or configuration errors in the macro when you apply it to a switch. In this example, the **ADDRESS** parameter value was not entered, the **snmp-server host** command failed, and the remainder of the macro is applied to the switch:

```
Device(config)# macro global trace snmp VALUE 7
Applying command...'snmp-server enable traps port-security'
Applying command...'snmp-server enable traps linkup'
Applying command...'snmp-server enable traps linkdown'
Applying command...'snmp-server host'
%Error Unknown error.
Applying command...'snmp-server ip precedence 7'
```

# macro global description

To enter a description about the macros that are applied to a switch, use the **macro global description** command in global configuration mode. Use the **no** form of this command to remove the description.

**macro global description** *text*

**no macro global description** *text*

## Syntax Description

| | |
|---|---|
| **description** *text* | Enters a description about the macros that are applied to the switch. |

## Command Default

This command has no default setting.

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use the **description** keyword to associate comment text or the macro name with a switch. When multiple macros are applied on a switch, the description text is from the last applied macro.

You can verify your settings by entering the **show parser macro description** command in privileged EXEC mode.

### Example

This example shows how to add a description to a switch:

```
Device(config)# macro global description udld aggressive mode enabled
```

# max-endpoints (coap-proxy configuration)

To specify the maximum number of endpoints that can be learnt on the device, use the **max-endpoints** command in coap-proxy configuration mode. To return to the default settings, use the **no** form of the command.

**max-endpoints** *number*
**no max-endpoints**

## Syntax Description

| | |
|---|---|
| *number* | Range is from 1 to 500 |

## Command Default

The default number of endpoints is 10.

## Command Modes

coap-proxy configuration (config-coap-proxy)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

To access coap-proxy configuration mode, enter the **coap proxy** command in global configuration mode.

### Example

This example shows how to specify maximum endpoints as 12 that can be learnt on the device.

```
Device(config)# coap proxy
Device(config-coap-proxy)# max-endpoints 12
```

# mdix auto

To enable the automatic medium-dependent interface crossover (auto-MDIX) feature on the interface, use the **mdix auto** command in interface configuration mode. To disable auto-MDIX, use the **no** form of this command.

**mdix auto**
**no mdix auto**

## Syntax Description

This command has no arguments or keywords.

## Command Default

Auto-MDIX is enabled.

## Command Modes

Interface configuration (config-if)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

When auto-MDIX is enabled, the interface automatically detects the required cable connection type (straight-through or crossover) and configures the connection appropriately.

When you enable auto-MDIX on an interface, you must also set the interface speed and duplex to **auto** so that the feature operates correctly.

When auto-MDIX (and autonegotiation of speed and duplex) is enabled on one or both of the connected interfaces, link up occurs, even if the cable type (straight-through or crossover) is incorrect.

Auto-MDIX is supported on all 10/100 and 10/100/1000 Mb/s interfaces and on 10/100/1000BASE-TX small form-factor pluggable (SFP) module interfaces. It is not supported on 1000BASE-SX or -LX SFP module interfaces.

You can verify the operational state of auto-MDIX on the interface by entering the **show controllers ethernet-controller** *interface-id* **phy** privileged EXEC command.

This example shows how to enable auto-MDIX on a port:

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# speed auto
Device(config-if)# duplex auto
Device(config-if)# mdix auto
Device(config-if)# end
```

# monitoring

To enable monitoring of all optical transceivers and to specify the time period for monitoring the transceivers, use the **monitoring** command in transceiver type configuration mode. To disable the monitoring, use the **no** form of this command.

**monitoring** [ **interval** *seconds* ]
**no monitoring** [**interval**]

## Syntax Description

| | |
|---|---|
| **interval** *seconds* | (Optional) Specifies the time interval for monitoring optical transceivers.<br>The range is from 300 to 3600 seconds, and the default interval time is 600 seconds. |

## Command Default

The interval time is 600 seconds.

## Command Modes

Transceiver type configuration (config-xcvr-type)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

You need digital optical monitoring (DOM) feature and transceiver module compatibility information to configure the **monitoring** command. Refer to the compatibility matrix to get the lists of Cisco platforms and minimum required software versions to support Gigabit Ethernet transceiver modules.

Gigabit Ethernet Transceivers transmit and receive Ethernet frames at a rate of a gigabit per second, as defined by the IEEE 802.3-2008 standard. Cisco's Gigabit Ethernet Transceiver modules support Ethernet applications across all Cisco switching and routing platforms. These pluggable transceivers offer a convenient and cost effective solution for the adoption in data center, campus, metropolitan area access and ring networks, and storage area networks.

The **interval** keyword enables you to change the default polling interval. For example, if you set the interval as 1500 seconds, polling happens at every 1500th second. During the polling period entSensorStatus of optical transceivers is set to *Unavailable*, and once the polling finishes entSensorStatus shows the actual status.

## Examples

This example shows how to enable monitoring of optical transceivers and set the interval time for monitoring to 1500 seconds:

```
Device# configure terminal
Device(config)# transceiver type all
Device(config-xcvr-type)# monitoring interval 1500
```

This example shows how to disable monitoring for all transceiver types:

```
Device(config-xcvr-type)# no monitoring
```

## Related Commands

| Command | Description |
|---|---|
| **transceiver type all** | Enables monitoring on all transceivers. |

# network-policy

To apply a network-policy profile to an interface, use the **network-policy** command in interface configuration mode. To remove the policy, use the **no** form of this command.

**network-policy** *profile-number*
**no network-policy**

## Syntax Description

| | |
|---|---|
| *profile-number* | The network-policy profile number to apply to the interface. |

## Command Default

No network-policy profiles are applied.

## Command Modes

Interface configuration (config-if)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use the **network-policy** *profile number* interface configuration command to apply a profile to an interface.

You cannot apply the **switchport voice vlan** command on an interface if you first configure a network-policy profile on it. However, if **switchport voice vlan** *vlan-id* is already configured on the interface, you can apply a network-policy profile on the interface. The interface then has the voice or voice-signaling VLAN network-policy profile applied.

This example shows how to apply network-policy profile 60 to an interface:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# network-policy 60
```

# network-policy profile (global configuration)

To create a network-policy profile and to enter network-policy configuration mode, use the **network-policy profile** command in global configuration mode. To delete the policy and to return to global configuration mode, use the **no** form of this command.

**network-policy profile** *profile-number*
**no network-policy profile** *profile-number*

**Syntax Description**

| | |
|---|---|
| *profile-number* | Network-policy profile number. The range is 1 to 4294967295. |

**Command Default**

No network-policy profiles are defined.

**Command Modes**

Global configuration (config)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

Use the **network-policy profile** global configuration command to create a profile and to enter network-policy profile configuration mode.

To return to privileged EXEC mode from the network-policy profile configuration mode, enter the **exit** command.

When you are in network-policy profile configuration mode, you can create the profile for voice and voice signaling by specifying the values for VLAN, class of service (CoS), differentiated services code point (DSCP), and tagging mode.

These profile attributes are contained in the Link Layer Discovery Protocol for Media Endpoint Devices (LLDP-MED) network-policy time-length-value (TLV).

This example shows how to create network-policy profile 60:

```
Device(config)# network-policy profile 60
Device(config-network-policy)#
```

# platform usb disable

To disable all the USB ports on a device, use the **platform usb disable** command in global configuration mode. To reenable all the USB ports on the device, use the **no platform usb disable** command.

**platform usb disable**
**no platform usb disable**

## Command Default

All the USB ports are enabled by default.

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Bengaluru 17.5.1 | This command was introduced. |

## Usage Guidelines

The **platform usb disable** command disables all the USB ports on both stacked and standalone devices, but not Bluetooth dongles connected to USB ports.

## Examples

The following example shows how to disable USB ports on a device:

```
Device> enable
Device# configure terminal
Device(config)# platform usb disable
This config cli may cause data corruption if there is some ongoing operation on usb device.
 Do you want to proceed [confirm]?
y
Device(config)# end
```

## Related Commands

| Command | Description |
|---|---|
| **show platform usb status** | Displays the status of the USB ports on a device. |

# port-dtls (coap-proxy configuration)

To configure a Datagram Transport Layer Security (DTLS) port, use the **port-dtls** command in coap-proxy configuration mode. To return to the default settings, use the **no** form of the command.

**port-dtls** *number*
**no port-dtls**

## Syntax Description

| | |
|---|---|
| *number* | Range is from 1 to 65000. |

## Command Default

The default port is 5683.

## Command Modes

coap-proxy configuration (config-coap-proxy)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

To access coap-proxy configuration mode, enter the **coap proxy** command in global configuration mode.

### Example

This example shows how to configure a dtls port .

```
Device(config)# coap proxy
Device(config-coap-proxy)# port-dtls 5899
```

# port-unsecure (coap-proxy configuration)

To configure a port, use the **port-unsecure** command in coap-proxy configuration mode. To return to the default settings, use the **no** form of the command.

**port-unsecure** *number*
**no port-dtls**

**Syntax Description**

| | |
|---|---|
| *number* | Range is from 1 to 65000. |

**Command Default**

The default port is 5683.

**Command Modes**

coap-proxy configuration (config-coap-proxy)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

To access coap-proxy configuration mode, enter the **coap proxy** command in global configuration mode.

### Example

This example shows how to configure a port .

```
Device(config)# coap proxy
Device(config-coap-proxy)# port-unsecure 5899
```

# power-priority

To configure Cisco StackPower power-priority values for a switch in a power stack and for its high-priority and low-priority PoE ports, use the **power-priority** command in switch stack-power configuration mode. To return to the default setting, use the **no** form of the command.

**power-priority** {**high** *value* | **low** *value* | **switch** *value*}
**no** **power-priority** {**high** | **low** | **switch**}

## Syntax Description

| | |
|---|---|
| **high** *value* | Sets the power priority for the ports configured as high-priority ports. The range is 1 to 27, with 1 as the highest priority. The **high** value must be lower than the value set for the low-priority ports and higher than the value set for the switch. |
| **low** *value* | Sets the power priority for the ports configured as low-priority ports. The range is 1 to 27. The **low** value must be higher than the value set for the high-priority ports and the value set for the switch. |
| **switch** *value* | Sets the power priority for the switch. The range is 1 to 27. The **switch** value must be lower than the values set for the low and high-priority ports. |

## Command Default

If no values are configured, the power stack randomly determines a default priority.

The default ranges are 1 to 9 for switches, 10 to 18 for high-priority ports, 19 to 27 for low-priority ports.

On non-PoE switches, the high and low values (for port priority) have no effect.

## Command Modes

Switch stack-power configuration (config-stack)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

To access switch stack-power configuration mode, enter the **stack-power switch** *switch-number* global configuration command.

Cisco StackPower power-priority values determine the order for shutting down switches and ports when power is lost and load shedding must occur. Priority values are from 1 to 27; the highest numbers are shut down first.

We recommend that you configure different priority values for each switch and for its high priority ports and low priority ports to limit the number of devices shut down at one time during a loss of power. If you try to configure the same priority value on different switches in a power stack, the configuration is allowed, but you receive a warning message.

**Note** This command is available only on switch stacks running the IP Base or IP Services feature set.

## Examples

This is an example of setting the power priority for switch 1 in power stack a to 7, for the high-priority ports to 11, and for the low-priority ports to 20.

```
Device(config)# stack-power switch 1
Device(config-switch-stackpower)# stack-id power_stack_a
Device(config-switch-stackpower)# power-priority high 11
Device(config-switch-stackpower)# power-priority low 20
Device(config-switch-stackpower)# power-priority switch 7
Device(config-switch-stackpower)# exit
```

# power inline

To configure the power management mode on Power over Ethernet (PoE) ports, use the **power inline** command in interface configuration mode. To return to the default settings, use the **no** form of this command.

**power inline** {**auto** [**max** *max-wattage*] | **never** | **port priority** {**high** | **low**} | **static** [**max** *max-wattage*]}
**no power inline** {**auto** | **never** | **port priority** {**high** | **low**} | **static** [**max** *max-wattage*]}

## Syntax Description

| | |
|---|---|
| **auto** | Enables powered-device detection. If enough power is available, automatically allocates power to the PoE port after device detection. Allocation is first-come, first-serve. |
| **max** *max-wattage* | (Optional) Limits the power allowed on the port. The range is 4000 to 30000 mW. If no value is specified, the maximum is allowed. |
| **never** | Disables device detection, and disables power to the port. |
| **port** | Configures the power priority of the port. The default priority is low. |
| **priority** {**high** \| **low**} | Sets the power priority of the port. In case of a power supply failure, ports configured as low priority are turned off first and ports configured as high priority are turned off last. The default priority is low. |
| **static** | Enables powered-device detection. Pre-allocates (reserves) power for a port before the switch discovers the powered device. This action guarantees that the device connected to the interface receives enough power. |

## Command Default

The default is **auto** (enabled).

The maximum wattage is 30,000 mW.

The default port priority is low.

## Command Default

Interface configuration (config-if)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

This command is supported only on PoE-capable ports. If you enter this command on a port that does not support PoE, this error message appears:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# power inline auto
                    ^
% Invalid input detected at '^' marker.
```

In a switch stack, this command is supported on all ports in the stack that support PoE.

Use the **max** *max-wattage* option to disallow higher-power powered devices. With this configuration, when the powered device sends Cisco Discovery Protocol (CDP) messages requesting more power than the maximum wattage, the switch removes power from the port. If the powered-device IEEE class maximum is greater than the maximum wattage, the switch does not power the device. The power is reclaimed into the global power budget.

**Note** The switch never powers any class 0 or class 3 device if the **power inline max max-wattage** command is configured for less than 30 W.

If the switch denies power to a powered device (the powered device requests more power through CDP messages or if the IEEE class maximum is greater than the maximum wattage), the PoE port is in a power-deny state. The switch generates a system message, and the Oper column in the **show power inline** privileged EXEC command output shows *power-deny*.

Use the **power inline static max** *max-wattage* command to give a port high priority. The switch allocates PoE to a port configured in static mode before allocating power to a port configured in auto mode. The switch reserves power for the static port when it is configured rather than upon device discovery. The switch reserves the power on a static port even when there is no connected device and whether or not the port is in a shutdown or in a no shutdown state. The switch allocates the configured maximum wattage to the port, and the amount is never adjusted through the IEEE class or by CDP messages from the powered device. Because power is pre-allocated, any powered device that uses less than or equal to the maximum wattage is guaranteed power when it is connected to a static port. However, if the powered device IEEE class is greater than the maximum wattage, the switch does not supply power to it. If the switch learns through CDP messages that the powered device needs more than the maximum wattage, the powered device is shut down.

If the switch cannot pre-allocate power when a port is in static mode (for example, because the entire power budget is already allocated to other auto or static ports), this message appears: Command rejected: power inline static: pwr not available. The port configuration remains unchanged.

When you configure a port by using the **power inline auto** or the **power inline static** interface configuration command, the port autonegotiates by using the configured speed and duplex settings. This is necessary to determine the power requirements of the connected device (whether or not it is a powered device). After the power requirements have been determined, the switch hardcodes the interface by using the configured speed and duplex settings without resetting the interface.

When you configure a port by using the **power inline never** command, the port reverts to the configured speed and duplex settings.

If a port has a Cisco powered device connected to it, you should not use the **power inline never** command to configure the port. A false link-up can occur, placing the port in an error-disabled state.

Use the **power inline port priority {high | low}** command to configure the power priority of a PoE port. Powered devices connected to ports with low port priority are shut down first in case of a power shortage.

You can verify your settings by entering the **show power inline** EXEC command.

## Examples

This example shows how to enable detection of a powered device and to automatically power a PoE port on a switch:

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# power inline auto
```

This example shows how to configure a PoE port on a switch to allow a class 1 or a class 2 powered device:

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# power inline auto max 7000
```

This example shows how to disable powered-device detection and to not power a PoE port on a switch:

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# power inline never
```

This example shows how to set the priority of a port to high, so that it would be one of the last ports to be shut down in case of power supply failure:

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# power inline port priority high
```

# power inline police

To enable policing of real-time power consumption on a powered device, use the **power inline police** command in interface configuration mode. To disable this feature, use the **no** form of this command

**power inline police** [**action** {**errdisable** | **log**}]
**no power inline police**

## Syntax Description

| | |
|---|---|
| **action errdisable** | (Optional) Configures the device to turn off power to the port if the real-time power consumption exceeds the maximum power allocation on the port. This is the default action. |
| **action log** | (Optional) Configures the device to generate a syslog message while still providing power to a connected device if the real-time power consumption exceeds the maximum power allocation on the port. |

## Command Default

Policing of the real-time power consumption of the powered device is disabled.

## Command Modes

Interface configuration (config-if)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

This command is supported only on Power over Ethernet (PoE)-capable ports. If you enter this command on a device or port that does not support PoE, an error message appears.

In a switch stack, this command is supported on all switches or ports in the stack that support PoE and real-time power-consumption monitoring.

When policing of the real-time power consumption is enabled, the device takes action when a powered device consumes more power than the allocated maximum amount.

When PoE is enabled, the device senses the real-time power consumption of the powered device. This feature is called *power monitoring* or *power sensing*. The device also polices the power usage with the *power policing* feature.

When power policing is enabled, the device uses one of the these values as the cutoff power on the PoE port in this order:

1. The user-defined power level that limits the power allowed on the port when you enter the **power inline auto max** *max-wattage* or the **power inline static max** *max-wattage* interface configuration command
2. The device automatically sets the power usage of the device by using CDP power negotiation or by the IEEE classification and LLPD power negotiation.

If you do not manually configure the cutoff-power value, the device automatically determines it by using CDP power negotiation or the device IEEE classification and LLDP power negotiation. If CDP or LLDP are not enabled, the default value of 30 W is applied. However without CDP or LLDP, the device does not allow devices to consume more than 15.4 W of power because values from 15400 to 30000 mW are only allocated based on CDP or LLDP requests. If a powered device consumes more than 15.4 W without CDP or LLDP negotiation, the device might be in violation of the maximum current *Imax* limitation and might experience an *Icut* fault for drawing more current than the maximum. The port remains in the fault state for a time before attempting to power on again. If the port continuously draws more than 15.4 W, the cycle repeats.

When a powered device connected to a PoE+ port restarts and sends a CDP or LLDP packet with a power TLV, the device locks to the power-negotiation protocol of that first packet and does not respond to power requests from the other protocol. For example, if the device is locked to CDP, it does not provide power to devices that send LLDP requests. If CDP is disabled after the device has locked on it, the device does not respond to LLDP power requests and can no longer power on any accessories. In this case, you should restart the powered device.

If power policing is enabled, the device polices power usage by comparing the real-time power consumption to the maximum power allocated on the PoE port. If the device uses more than the maximum power allocation (or *cutoff power*) on the port, the device either turns power off to the port, or the device generates a syslog message and updates the LEDs (the port LEDs are blinking amber) while still providing power to the device.

- To configure the device to turn off power to the port and put the port in the error-disabled state, use the **power inline police** interface configuration command.
- To configure the device to generate a syslog message while still providing power to the device, use the **power inline police action log** command.

If you do not enter the **action log** keywords, the default action is to shut down the port, turn off power to it, and put the port in the PoE error-disabled state. To configure the PoE port to automatically recover from the error-disabled state, use the **errdisable detect cause inline-power** global configuration command to enable error-disabled detection for the PoE cause and the **errdisable recovery cause inline-power interval** *interval* global configuration command to enable the recovery timer for the PoE error-disabled cause.

**Caution** If policing is disabled, no action occurs when the powered device consumes more than the maximum power allocation on the port, which could adversely affect the device.

You can verify your settings by entering the **show power inline police** privileged EXEC command.

## Examples

This example shows how to enable policing of the power consumption and configuring the device to generate a syslog message on the PoE port on a device:

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# power inline police action log
```

# power supply

To configure and manage the internal power supplies on a switch, use the **power supply** command in privileged EXEC mode.

**power supply** *stack-member-number* **slot** {**A** | **B**} {**off** | **on**}

## Syntax Description

| | |
|---|---|
| *stack-member-number* | Stack member number for which to configure the internal power supplies. The range is 1 to 9, depending on the number of switches in the stack.<br>This parameter is available only on stacking-capable switches. |
| **slot** | Selects the switch power supply to set. |
| **A** | Selects the power supply in slot A. |
| **B** | Selects the power supply in slot B.<br>**Note** Power supply slot B is the closest slot to the outer edge of the switch. |
| **off** | Sets the switch power supply to off. |
| **on** | Sets the switch power supply to on. |

## Command Default

The switch power supply is on.

## Command Modes

Privileged EXEC

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

The **power supply** command applies to a switch or to a switch stack where all switches are the same platform.

In a switch stack with the same platform switches, you must specify the stack member before entering the **slot** {**A** | **B**} **off** or **on** keywords.

To return to the default setting, use the **power supply** *stack-member-number* **on** command.

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

## Examples

This example shows how to set the power supply in slot A to off:

```
Device> power supply 2 slot A off
Disabling Power supply A may result in a power loss to PoE devices and/or switches ...
Continue? (yes/[no]): yes
Device
Jun 10 04:52:54.389: %PLATFORM_ENV-6-FRU_PS_OIR: FRU Power Supply 1 powered off
Jun 10 04:52:56.717: %PLATFORM_ENV-1-FAN_NOT_PRESENT: Fan is not present
```

This example shows how to set the power supply in slot A to on:

```
Device> power supply 1 slot B on
Jun 10 04:54:39.600: %PLATFORM_ENV-6-FRU_PS_OIR: FRU Power Supply 1 powered on
```

This example shows the output of the show env power command:

```
Device> show env power
SW  PID                 Serial#     Status           Sys Pwr  PoE Pwr  Watts
--  ------------------  ----------  ---------------  -------  -------  -----
1A  PWR-1RUC2-640WAC    DCB1705B05B OK               Good     Good     250/390
1B  Not Present
```

# power supply autoLC shutdown

To enable automatic shutdown control on linecards, use the **power supply autoLC shutdown** command in global configuration mode. This command is enabled by default and cannot be disabled. The `AutoLC shutdown cannot be disabled` message will be displayed if you try to disable it.

**power supply autoLC shutdown**
**no power supply autoLC shutdown**

## Syntax Description

This command has no arguments or keywords.

## Command Default

Automatic shutdown control on linecards is enabled.

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Examples

This example shows how to enable automatic shutdown on linecards:

```
Device> enable
Device# configure terminal
Device(config)# power supply autoLC shutdown
```

# resource directory (coap-proxy configuration)

To unicast upstream resource directory server to which the switch can act as a COAP client, use the **resource directory** command in coap-proxy configuration mode. To return to the default settings, use the **no** form of the command.

A maximum of five ip-lists can be configured, for each ipv4 or ipv6, using the resource directory command.

**resource directory** {**ipv4** | **ipv6**}[*ip-address*]
**no resource directory**

**Syntax Description**

| | |
|---|---|
| **ipv4** *ip-address* | Specifies IPv4 address. |
| **ipv6** *ip-address* | Specifies IPv6 address. |

**Command Modes**

coap-proxy configuration (config-coap-proxy)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

To access coap-proxy configuration mode, enter the **coap proxy** command in global configuration mode.

### Example

This example shows how to unicast upstream resource directory server to which the switch can act as a COAP client.

```
Device(config)# coap proxy
Device(config-coap-proxy)# resource-directory ipv4 192.168.1.1
```

# security (coap-proxy configuration)

To configure CoAP security features, use the **security** command in coap-proxy configuration mode. To return to the default settings, use the **no** form of the command.

**security** {**none** [{**ipv4** { *ip-address ip-mask/prefix*} | **ipv6**{ *ip-address ip-mask/prefix*} | **list**{*ipv4-list-name ipv6-list-name*}}] | **dtls** {[**id-trustpoint** {*identity-trustpoint label*}][**verification-trustpoint** { *verification-trustpoint*}] | [{**ipv4** { *ip-address ip-mask/prefix*} | **ipv6**{ *ip-address ip-mask/prefix*} | **list**{*ipv4-list-name ipv6-list-name*}}]}}
**no security**

## Syntax Description

| | |
|---|---|
| **none** | Indicates no security on that port.<br>**Note** A maximum of five ipv4 and five ipv6 addresses can be associated. |
| **dtls** | The DTLS security takes RSA trustpoint and Verification trustpoint which are optional. Without 1.1.0.0 255.255.0.0 Verification trustpoint it does the normal Public Key Exchange.<br>**Note** A maximum of five ipv4 and five ipv6 addresses can be associated. |

## Command Modes

coap-proxy configuration (config-coap-proxy)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

To access coap-proxy configuration mode, enter the **coap proxy** command in global configuration mode.

### Example

This example shows how to configure no security on the port.

```
Device(config)# coap proxy
Device(config-coap-proxy)# security none ipv4 1.1.0.0 255.255.0.0
```

# shell trigger

To create an event trigger, use the **shell trigger** command in global configuration mode. Use the **no** form of this command to delete the trigger.

**shell trigger** *identifier description*

**no shell trigger** *identifier description*

## Syntax Description

| | |
|---|---|
| *identifier* | Specifies the event trigger identifier. The identifier should have no spaces or hyphens between words. |
| *description* | Specifies the event trigger description text. |

## Command Default

System-defined event triggers:

- CISCO_DMP_EVENT
- CISCO_IPVSC_AUTO_EVENT
- CISCO_PHONE_EVENT
- CISCO_SWITCH_EVENT
- CISCO_ROUTER_EVENT
- CISCO_WIRELESS_AP_EVENT
- CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use this command to create user-defined event triggers for use with the **macro auto device** and the **macro auto execute** commands.

To support dynamic device discovery when using IEEE 802.1x authentication, you need to configure the RADIUS authentication server to support the Cisco attribute-value pair: **auto-smart-port**=*event trigger*.

### Example

This example shows how to create a user-defined event trigger called RADIUS_MAB_EVENT:

```
Device(config)# shell trigger RADIUS_MAB_EVENT MAC_AuthBypass Event
Device(config)# end
```

# show beacon all

To display the status of beacon LED on the device, use the **show beacon all** command in privileged EXEC mode.

**show beacon** {**rp** {**active** | **standby**} | **slot** *slot-number* } | **all**}

## Syntax Description

| | |
|---|---|
| **rp** {**active** \| **standby**} | Specifies the active or the standby Switch whose beacon LED status is to be displayed. |
| **slot** *slot-num* | Specifies the slot whose beacon LED status is to be displayed. |
| all | Displays the status of all beacon LEDs. |

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Command Default

This command has no default settings.

## Command Modes

Privileged EXEC (#)

## Usage Guidelines

Use the command **show beacon all** to know the status of all beacon LEDs.

### Sample output of *show beacon all* command.

```
Device#show beacon all
Switch#   Beacon Status
-----------------------
*1        OFF
```

### Sample output of *show beacon rp* command.

```
Device#show beacon rp active
Switch#   Beacon Status
-----------------------
*1        OFF
```

```
Device#show beacon slot 1
Switch#   Beacon Status
-----------------------
*1        OFF
```

# show coap dtls endpoints

To display the CoAP dtls endpoints, use the **show coap dtls endpoints** command in user EXEC or privileged EXEC mode.

**show coap dtls endpoints**

**Command Default** This command has no arguments or keywords.

**Command Modes** User EXEC (>)

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

### Example

This example shows how to display the CoAP dtls endpoint:

```
Device# show coap dtls endpoints
#     Index StateString StateValue  Port IP
---------------------------------------------------------------
```

# show coap endpoints

To display the CoAP endpoints, use the **show coap endpoints** command in user EXEC or privileged EXEC mode.

**show coap endpoints**

**Command Default** This command has no arguments or keywords.

**Command Modes** User EXEC (>)

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

### Example

This example shows how to display the CoAP endpoint

```
Device# show coap endpoints
List of all endpoints :

Code : D - Discovered , N - New
#    Status  Age(s)    LastWKC(s)    IP
---------------------------------------------------------------

Endpoints - Total : 0 Discovered : 0 New : 0
```

# show coap globals

To display the CoAP globals, use the **show coap globals** command in user EXEC or privileged EXEC mode.

**show coap globals**

**Command Default**

This command has no arguments or keywords.

**Command Modes**

User EXEC (>)

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

### Example

The following is sample output from the **show coap globals** command:

This example shows how to display the CoAP configuration:

```
Device# show coap dtls globals
Coap System Timer Values :
 Discovery  : 120 sec
 Cache Exp  : 5 sec
 Keep Alive : 120 sec
 Client DB  : 5 sec
 Query Queue: 500 ms
 Ack delay  : 500 ms
 Timeout    : 5 sec
 Ageout     : 300 sec

Max Endpoints      : 10

Max DTLS Endpoints : 20
Resource Disc Mode : POST
```

# show coap resources

To display the CoAP resources, use the **show coap resources** command in user EXEC or privileged EXEC mode.

**show coap resources**

### Command Default

This command has no arguments or keywords.

### Command Modes

User EXEC (>)

Privileged EXEC (#)

### Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

### Example

This example shows how to display the CoAP resources:

```
Device# show coap resources
Link format data =

</>
</cisco/flood>
</cisco/context>
</cisco/showtech>
</cisco/discover>
</cisco/sleep>
</cisco/lldp>
```

# show coap stats

To display the CoAP stats, use the **show coap stats** command in user EXEC or privileged EXEC mode.

**show coap stats**

**Command Default** This command has no arguments or keywords.

**Command Modes** User EXEC (>)

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

### Example

This example shows how to display the CoAP stats:

```
Device# show coap stats
Coap Stats :
Endpoints    : 0
Requests     : 20
Ext Queries  : 0
New Endpoints: 0
```

# show coap version

To display the CoAP version, use the **show coap version** command in user EXEC or privileged EXEC mode.

**show coap version**

**Command Default**

This command has no arguments or keywords.

**Command Modes**

User EXEC (>)

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

### Example

This example shows how to display the CoAP version:

```
Device# show coap version
CoAP version 1.0.5
RFC 7252
```

# show device classifier attached

To display the devices connected to a switch and their associated properties, use the **show device classifier attached** command in user EXEC mode.

**show device classifier attached** [{**detail** | **interface** *interface_id* | **mac-address** *mac_address*}]

## Syntax Description

| | |
|---|---|
| **detail** | Displays detailed device classifier information. |
| **interface** *interface_id* | Displays information about devices attached to the specified interface. |
| **mac** *mac_address* | Displays device information for the specified endpoint. |

## Command Modes

User EXEC (>)

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use this command to display the devices connected to a switch. Use the **show device classifier attached** command in privileged EXEC mode to display the configurable parameters for a device.

### Example

This example shows how to use the **show device classifier attached** command with no optional keywords to view the devices connected to the switch:

```
Device# show device classifier attached
MAC_Address       Port_Id     Profile Name
==============    =======     ============================
000a.b8c6.1e07    Gi1/0/2    Cisco-Device
001f.9e90.1250    Gi1/0/4    Cisco-AP-Aironet-1130
======================================================
```

This example shows how to use the **show device classifier attached** command in privileged EXEC mode with the optional **mac-address** keyword to view summary information about the connected device with the specified MAC address:

```
Device# show device classifier attached mac-address 001f.9e90.1250
MAC_Address       Port_Id     Profile Name
==============    =======     ============================
001f.9e90.1250    Gi1/0/4    Cisco-AP-Aironet-1130
==========================================================
```

This example shows how to use the **show device classifier attached** command in privileged EXEC mode with the optional **mac-address** and **detail** keywords to view detailed information about the connected device with the specified MAC address:

```
Device# show device classifier attached mac-address 001f.9e90.1250 detail
MAC_Address       Port_Id     Certainty Parent    ProfileType      Profile Name
     Device_Name
==============    =======     ========= ======    ===========   ============================
    ===========
001f.9e90.1250    Gi1/0/4      40       2         Built-in       Cisco-AP-Aironet-1130
    cisco AIR-LAP1131AG-E-K9
==========================================================================================
```

This example shows how to use the **show device classifier attached** command in privileged EXEC mode with the optional **interface** keyword to view summary information about the device connected to the specified interface:

```
Device# show device classifier attached interface gi 1/0/2
MAC_Address       Port_Id          Profile Name
==============    =======     ============================
000a.b8c6.1e07    Gi1/0/2    Cisco-Device
=====================================================
```

This example shows how to use the **show device classifier attached** command in privileged EXEC mode with the optional **interface** and **detail** keywords to view detailed information about the device connected to the specified interface:

```
Device# show device classifier attached interface gi 1/0/2 detail
MAC_Address       Port_Id     Certainty Parent    ProfileType  Profile Name
     Device_Name
==============    =======     ========= ======    ===========   ============================
    ===========
000a.b8c6.1e07    Gi1/0/2      10       0        Default       Cisco-Device        cisco
WS-C2960-48TT-L
==========================================================================================
```

# show device classifier clients

To display the clients using the device classifier facility on the switch, use the **show device classifier clients** command in user EXEC mode.

**show device classifier clients**

## Command Default

This command has no arguments or keywords.

## Command Modes

User EXEC (>)

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Device classifier (DC) is enabled by default when you enable a client application (for example, Auto SmartPorts) that uses its functionality. Use the **show device classifier clients** command to display the clients that are using the DC feature on the switch.

As long as any clients are using the DC, you cannot disable it by using the **no device classifier** command. If you attempt to disable the DC while a client is using it, an error message appears.

### Example

This example shows how to use the **show device classifier clients** command to view the clients using the DC on the switch:

```
Device# show device classifier clients
Client Name
====================
Auto Smart Ports

This example shows the error message that appears when you attempt to disable DC while a
client is using it:
Switch(config)# no device classifier
These subsystems should be disabled before disabling Device classifier
Auto Smart Ports

% Error - device classifier is not disabled
```

# show device classifier profile type

To display all the device types recognized by the device classifier, use the **show device classifier profile type** command in user EXEC mode.

**show device classifier profile type** [{**table** [{*built-in default*}] | **string** *filter_string*}]

## Syntax Description

| | |
|---|---|
| **table** | Displays device classification in a table. |
| *built-in* | Displays device classification information from the built-in device table. |
| *default* | Displays device classification information from the default device table. |
| **filter** *string* | Displays information for devices that match the filter. |

## Command Modes

User EXEC (>)

Privileged EXEC (#)

## Command History

| **Release** | **Modification** |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

This command displays all the device types recognized by the device classification engine. The number of available device types is the number of profiles stored on the switch. Because the number of profiles can be very large, you can use the **filter** keyword to limit the command output.

### Example

This example shows how to use the **show device classifier profile type** command in privileged EXEC mode with no optional keywords to view the devices recognized by the device classifier:

```
Device# show device classifier profile type table
  Valid         Type       Profile Name                         min Conf   ID
===========  =========  ==================                      ========  ====
  Valid        Default      Apple-Device                             10       0
  Valid        Default      Aruba-Device                             10       1
  Valid        Default      Avaya-Device                             10       2
  Valid        Default      Avaya-IP-Phone                           20       3
  Valid        Default      BlackBerry                               20       4
  Valid        Default      Cisco-Device                             10       5
  Valid        Default      Cisco-IP-Phone                           20       6
  Valid        Default      Cisco-IP-Phone-7902                      70       7
  Valid        Default      Cisco-IP-Phone-7905                      70       8
  Valid        Default      Cisco-IP-Phone-7906                      70       9
  Valid        Default      Cisco-IP-Phone-7910                      70       10
  Valid        Default      Cisco-IP-Phone-7911                      70       11
  Valid        Default      Cisco-IP-Phone-7912                      70       12
  Valid        Default      Cisco-IP-Phone-7940                      70       13
  Valid        Default      Cisco-IP-Phone-7941                      70       14
  Valid        Default      Cisco-IP-Phone-7942                      70       15
Valid        Default      Cisco-IP-Phone-7945                        70         16
Valid        Default      Cisco-IP-Phone-7945G                       70         17
Valid        Default      Cisco-IP-Phone-7960                        70         18
Valid        Default      Cisco-IP-Phone-7961                        70         19
Valid        Default      Cisco-IP-Phone-7962                        70         20
Valid        Default      Cisco-IP-Phone-7965                        70         21
Valid        Default      Cisco-IP-Phone-7970                        70         22
Valid        Default      Cisco-IP-Phone-7971                        70         23
Valid        Default      Cisco-IP-Phone-7975                        70         24
Valid        Default      Cisco-IP-Phone-7985                        70         25
Valid        Default      Cisco-IP-Phone-9971                        70         26
Valid        Default      Cisco-WLC-2100-Series                      40         27
Valid        Default      DLink-Device                               10         28
Valid        Default      Enterasys-Device                           10         29
Valid        Default      HP-Device                                  10         30
Valid        Default      HP-JetDirect-Printer                       30         31
Valid        Default      Lexmark-Device                             10         32
Valid        Default      Lexmark-Printer-E260dn                     30         33
Valid        Default      Microsoft-Device                           10         34
Valid        Default      Netgear-Device                             10         35
Valid        Default      NintendoWII                                10         36
Valid        Default      Nortel-Device                              10         37
Valid        Default      Nortel-IP-Phone-2000-Series                20         38
Valid        Default      SonyPS3                                    10         39
Valid        Default      XBOX360                                    20         40
Valid        Default      Xerox-Device                               10         41
Valid        Default      Xerox-Printer-Phaser3250                   30         42
Valid        Default      Aruba-AP                                   20         43
Valid        Default      Cisco-Access-Point                         10         44
Valid        Default      Cisco-IP-Conference-Station-7935           70         45
Valid        Default      Cisco-IP-Conference-Station-7936           70         46
Valid        Default      Cisco-IP-Conference-Station-7937           70         47
Valid        Default      DLink-DAP-1522                             20         48
Valid        Default      Cisco-AP-Aironet-1130                      30         49
Valid        Default      Cisco-AP-Aironet-1240                      30         50
Valid        Default      Cisco-AP-Aironet-1250                      30         51
Valid        Default      Cisco-AIR-LAP                              25         52
Valid        Default      Cisco-AIR-LAP-1130                         30         53
Valid        Default      Cisco-AIR-LAP-1240                         50         54
Valid        Default      Cisco-AIR-LAP-1250                         50         55
Valid        Default      Cisco-AIR-AP                               25         56
Valid        Default      Cisco-AIR-AP-1130                          30         57
Valid        Default      Cisco-AIR-AP-1240                          50         58
Valid        Default      Cisco-AIR-AP-1250                          50         59
Invalid      Default      Sun-Workstation                            10         60
Valid        Default      Linksys-Device                             20         61
Valid        Default      LinksysWAP54G-Device                       30         62
Valid        Default      HTC-Device                                 10         63
Valid        Default      MotorolaMobile-Device                      10         64
Valid        Default      VMWare-Device                              10         65
Valid        Default      ISE-Appliance                              10         66
Valid        Built-in     Cisco-Device                               10         0
Valid        Built-in     Cisco-Router                               10         1
Valid        Built-in     Router                                     10         2
Valid        Built-in     Cisco-IP-Camera                            10         3
Valid        Built-in     Cisco-IP-Camera-2xxx                       30         4
Valid        Built-in     Cisco-IP-Camera-2421                       50         5
Valid        Built-in     Cisco-IP-Camera-2500                       50         6
Valid        Built-in     Cisco-IP-Camera-2520                       50         7
Valid        Built-in     Cisco-IP-Camera-2530                       50         8
Valid        Built-in     Cisco-IP-Camera-4xxx                       50         9
Valid        Built-in     Cisco-Transparent-Bridge                   8          10
Valid        Built-in     Transparent-Bridge                         8          11
Valid        Built-in     Cisco-Source-Bridge                        10         12
Valid        Built-in     Cisco-Switch                                    10        13
Valid        Built-in     Cisco-IP-Phone                                  20        14
Valid        Built-in     IP-Phone                                        20        15
Valid        Built-in     Cisco-DMP                                       10        16
Valid        Built-in     Cisco-DMP-4305G                                 70        17
Valid        Built-in     Cisco-DMP-4310G                                 70        18
Valid        Built-in     Cisco-DMP-4400G                                 70        19
Valid        Built-in     Cisco-WLC-2100-Series                           40        20
Valid        Built-in     Cisco-Access-Point                              10        21
Valid        Built-in     Cisco-AIR-LAP                                   30        22
Valid        Built-in     Cisco-AIR-AP                                    30        23
Valid        Built-in     Linksys-Device                                  20        24
```

# show environment

To display fan, temperature, and power information, use the **show environment** command in EXEC mode.

**show environment** { **all** | **fan** | **power** | **stack** | **temperature** }

## Syntax Description

| | |
|---|---|
| **all** | Displays the fan and temperature environmental status and the status of the internal power supplies. |
| **fan** | Displays the switch fan status. |
| **power** | Displays the internal power status of the active switch. |
| **stack** | Displays all environmental status for each switch in the stack or for the specified switch.<br>This keyword is available only on stacking-capable switches. |
| **temperature** | Displays the switch temperature status. |

## Command Default

None

## Command Modes

User EXEC (>)

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use the **show environment** EXEC command to display the information for the switch being accessed—a standalone switch or the active switch. Use this command with the **stack** keyword to display all information for the stack or for the specified stack member.

If you enter the **show environment temperature status** command, the command output shows the switch temperature state and the threshold level.

You can also use the **show environment temperature** command to display the switch temperature status. The command output shows the green and yellow states as *OK* and the red state as *FAULTY*.

On the C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, the **show environment temperature** command does not display the correct value of 74 for yellow threshold system temperature if the device is upgraded from an older release where the supported value is 71. To fix this, run the **no system environment temperature threshold yellow** command.

## Examples

This example shows a sample output of the **show environment all** command:

```
Device> show environment all

Switch 1 FAN 1 is OK
Switch 1 FAN 2 is OK
Switch 1 FAN 3 is OK
FAN PS-1 is NOT PRESENT
FAN PS-2 is OK
Switch 1: SYSTEM TEMPERATURE is OK
Inlet Temperature Value: 25 Degree Celsius
Temperature State: GREEN
Yellow Threshold : 46 Degree Celsius
Red Threshold    : 56 Degree Celsius

Hotspot Temperature Value: 35 Degree Celsius
Temperature State: GREEN
Yellow Threshold : 105 Degree Celsius
Red Threshold    : 125 Degree Celsius
SW  PID                 Serial#     Status           Sys Pwr  PoE Pwr  Watts
--  ------------------  ----------  ---------------  -------  -------  -----
1A  Unknown             Unknown      No Input Power  Bad      Bad      235
1B  PWR-C1-350WAC       DCB2137H04P  OK              Good     Good     350
```

This example shows a sample output of the **show environment power** command:

```
Device> show environment power

SW  PID                 Serial#     Status           Sys Pwr  PoE Pwr  Watts
--  ------------------  ----------  ---------------  -------  -------  -----
1A  Unknown             Unknown      No Input Power  Bad      Bad      235
1B  PWR-C1-350WAC       DCB2137H04P  OK              Good     Good     350
```

This example shows a sample output of the **show environment stack** command:

```
Device# show environment stack

System Temperature Value: 41 Degree Celsius
System Temperature State: GREEN
Yellow Threshold : 66 Degree Celsius
Red Threshold    : 76 Degree Celsius
```

This example shows a sample output of the **show environment temperature** command:

```
Device> show environment temperature

Switch 1: SYSTEM TEMPERATURE is OK
Inlet Temperature Value: 25 Degree Celsius
Temperature State: GREEN
Yellow Threshold : 46 Degree Celsius
Red Threshold    : 56 Degree Celsius

Hotspot Temperature Value: 35 Degree Celsius
Temperature State: GREEN
Yellow Threshold : 105 Degree Celsius
Red Threshold    : 125 Degree Celsius
```

***Table 3: States in the show environment temperature status Command Output***

| State | Description |
|---|---|
| Green | The switch temperature is in the *normal* operating range. |
| Yellow | The temperature is in the *warning* range. You should check the external temperature around the switch. |
| Red | The temperature is in the *critical* range. The switch might not run properly if the temperature is in this range. |

# show errdisable detect

To display error-disabled detection status, use the **show errdisable detect** command in EXEC mode.

**show errdisable detect**

**Syntax Description**

This command has no arguments or keywords.

**Command Default**

None

**Command Modes**

User EXEC (>)

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

A gbic-invalid error reason refers to an invalid small form-factor pluggable (SFP) module.

The error-disable reasons in the command output are listed in alphabetical order. The mode column shows how error-disable is configured for each feature.

You can configure error-disabled detection in these modes:

- port mode—The entire physical port is error-disabled if a violation occurs.
- vlan mode—The VLAN is error-disabled if a violation occurs.
- port/vlan mode—The entire physical port is error-disabled on some ports and is per-VLAN error-disabled on other ports.

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

```
Device> show errdisable detect
ErrDisable Reason    Detection    Mode
-----------------    ---------    ----
arp-inspection       Enabled      port
bpduguard            Enabled      vlan
channel-misconfig    Enabled      port
community-limit      Enabled      port
dhcp-rate-limit      Enabled      port
dtp-flap             Enabled      port
gbic-invalid         Enabled      port
inline-power         Enabled      port
invalid-policy       Enabled      port
l2ptguard            Enabled      port
link-flap            Enabled      port
loopback             Enabled      port
lsgroup              Enabled      port
pagp-flap            Enabled      port
psecure-violation    Enabled      port/vlan
security-violatio    Enabled      port
sfp-config-mismat    Enabled      port
storm-control        Enabled      port
```

```
udld                    Enabled      port
vmps                    Enabled      port
```

# show errdisable recovery

To display the error-disabled recovery timer information, use the **show errdisable recovery** command in EXEC mode.

**show errdisable recovery**

## Syntax Description

This command has no arguments or keywords.

## Command Default

None

## Command Modes

User EXEC (>)

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

A gbic-invalid error-disable reason refers to an invalid small form-factor pluggable (SFP) module interface.

**Note** Though visible in the output, the unicast-flood field is not valid.

# show idprom tan

To display the Identification Programmable Read-Only Memory (IDPROM) top assembly part number and revision number, use the **show idprom tan** command in privileged EXEC mode.

**show idprom tan** [ **switch** [*switch-num*] ]

## Syntax Description

This command has no arguments or keywords.

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Dublin 17.12.1 | This command was introduced. |

## Examples

The following example shows how to display the IDPROM top assembly part number and revision number:

```
Device# show idprom tan switch 1

Switch 01
--------------------------------------------------------------
Top Assembly Part Number and Revision Number for Active Switch
--------------------------------------------------------------
Top Assy. Part Number           : 68-101751-01
Top Assy. Revision Number       : E0
```

# show ip interface

To display the usability status of interfaces configured for IP, use the **show ip interface** command in privileged EXEC mode.

**show ip interface** [*type number*] [**brief**]

## Syntax Description

| | |
|---|---|
| *type* | (Optional) Interface type. |
| *number* | (Optional) Interface number. |
| **brief** | (Optional) Displays a summary of the usability status information for each interface.<br>**Note** The output of the **show ip interface brief** command displays information of all the available interfaces whether or not the corresponding network module for these interfaces are connected. These interfaces can be configured if the network module is connected. Run the **show interface status** command to see which network modules are connected. |

## Command Default

The full usability status is displayed for all interfaces configured for IP.

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

The Cisco IOS software automatically enters a directly connected route in the routing table if the interface is usable (which means that it can send and receive packets). If an interface is not usable, the directly connected routing entry is removed from the routing table. Removing the entry lets the software use dynamic routing protocols to determine backup routes to the network, if any.

If the interface can provide two-way communication, the line protocol is marked "up." If the interface hardware is usable, the interface is marked "up."

If you specify an optional interface type, information for that specific interface is displayed. If you specify no optional arguments, information on all the interfaces is displayed.

When an asynchronous interface is encapsulated with PPP or Serial Line Internet Protocol (SLIP), IP fast switching is enabled. A **show ip interface** command on an asynchronous interface encapsulated with PPP or SLIP displays a message indicating that IP fast switching is enabled.

You can use the **show ip interface brief** command to display a summary of the device interfaces. This command displays the IP address, the interface status, and other information.

The **show ip interface brief** command does not display any information related to Unicast RPF.

## Examples

The following example shows interface information on Gigabit Ethernet interface 1/0/1:

```
Device# show ip interface gigabitethernet 1/0/1
```

```
GigabitEthernet1/0/1 is up, line protocol is up
  Internet address is 10.1.1.1/16
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP Feature Fast switching turbo vector
  IP VPN Flow CEF switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Policy routing is enabled, using route map PBR
  Network address translation is disabled
  BGP Policy Mapping is disabled
  IP Multi-Processor Forwarding is enabled
     IP Input features, "PBR",
         are not supported by MPF and are IGNORED
     IP Output features, "NetFlow",
         are not supported by MPF and are IGNORED
```

The following example shows how to display the usability status for a specific VLAN:

```
Device# show ip interface vlan 1

Vlan1 is up, line protocol is up
  Internet address is 10.0.0.4/24
  Broadcast address is 255.255.255.255
Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP Fast switching turbo vector
  IP Normal CEF switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect inbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
  Sampled Netflow is disabled
  IP multicast multilayer switching is disabled
  Netflow Data Export (hardware) is enabled
```

The table below describes the significant fields shown in the display.

***Table 4: show ip interface Field Descriptions***

| Field | Description |
|---|---|
| Broadcast address is | Broadcast address. |
| Peer address is | Peer address. |
| MTU is | MTU value set on the interface, in bytes. |
| Helper address | Helper address, if one is set. |
| Directed broadcast forwarding | Shows whether directed broadcast forwarding is enabled. |
| Outgoing access list | Shows whether the interface has an outgoing access list set. |
| Inbound access list | Shows whether the interface has an incoming access list set. |
| Proxy ARP | Shows whether Proxy Address Resolution Protocol (ARP) is enabled for the interface. |
| Security level | IP Security Option (IPSO) security level set for this interface. |
| Split horizon | Shows whether split horizon is enabled. |
| ICMP redirects | Shows whether redirect messages will be sent on this interface. |
| ICMP unreachables | Shows whether unreachable messages will be sent on this interface. |
| ICMP mask replies | Shows whether mask replies will be sent on this interface. |
| IP fast switching | Shows whether fast switching is enabled for this interface. It is generally enabled on serial interfaces, such as this one. |
| IP Flow switching | Shows whether Flow switching is enabled for this interface. |

| Field | Description |
|---|---|
| IP CEF switching | Shows whether Cisco Express Forwarding switching is enabled for the interface. |
| IP multicast fast switching | Shows whether multicast fast switching is enabled for the interface. |
| IP route-cache flags are Fast | Shows whether NetFlow is enabled on an interface. Displays "Flow init" to specify that NetFlow is enabled on the interface. Displays "Ingress Flow" to specify that NetFlow is enabled on a subinterface using the **ip flow ingress**command. Shows "Flow" to specify that NetFlow is enabled on a main interface using the **ip route-cache flow** command. |
| Router Discovery | Shows whether the discovery process is enabled for this interface. It is generally disabled on serial interfaces. |
| IP output packet accounting | Shows whether IP accounting is enabled for this interface and what the threshold (maximum number of entries) is. |
| TCP/IP header compression | Shows whether compression is enabled. |
| WCCP Redirect outbound is disabled | Shows the status of whether packets received on an interface are redirected to a cache engine. Displays "enabled" or "disabled." |
| WCCP Redirect exclude is disabled | Shows the status of whether packets targeted for an interface will be excluded from being redirected to a cache engine. Displays "enabled" or "disabled." |
| Netflow Data Export (hardware) is enabled | NetFlow Data Expert (NDE) hardware flow status on the interface. |

The following example shows how to display a summary of the usability status information for each interface:

```
Device# show ip interface brief

Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0     unassigned      YES NVRAM  down                  down
GigabitEthernet1/0/1   unassigned      YES NVRAM  down                  down
GigabitEthernet1/0/2   unassigned      YES unset  down                  down
GigabitEthernet1/0/3   unassigned      YES unset  down                  down
GigabitEthernet1/0/4   unassigned      YES unset  down                  down
GigabitEthernet1/0/5   unassigned      YES unset  down                  down
GigabitEthernet1/0/6   unassigned      YES unset  down                  down
GigabitEthernet1/0/7   unassigned      YES unset  down                  down

<output truncated>
```

*Table 5: show ip interface brief Field Descriptions*

| Field | Description |
|---|---|
| Interface | Type of interface. |

| Field | Description |
|---|---|
| IP-Address | IP address assigned to the interface. |
| OK? | "Yes" means that the IP Address is valid. "No" means that the IP Address is not valid. |
| Method | The Method field has the following possible values:<br>• RARP or SLARP: Reverse Address Resolution Protocol (RARP) or Serial Line Address Resolution Protocol (SLARP) request.<br>• BOOTP: Bootstrap protocol.<br>• TFTP: Configuration file obtained from the TFTP server.<br>• manual: Manually changed by the command-line interface.<br>• NVRAM: Configuration file in NVRAM.<br>• IPCP: **ip address negotiated** command.<br>• DHCP: **ip address dhcp** command.<br>• unset: Unset.<br>• other: Unknown. |
| Status | Shows the status of the interface. Valid values and their meanings are:<br>• up: Interface is up.<br>• down: Interface is down.<br>• administratively down: Interface is administratively down. |
| Protocol | Shows the operational status of the routing protocol on this interface. |

## Related Commands

| Command | Description |
|---|---|
| **ip interface** | Configures a virtual gateway IP interface on a Secure Socket Layer Virtual Private Network (SSL VPN) gateway |
| **show interface status** | Displays the status of the interface. |

# show interfaces

To display the administrative and operational status of all interfaces or for a specified interface, use the **show interfaces** command in the EXEC mode.

**show interfaces** [{ *interface-id* | **vlan** *vlan-id* }] [{ **accounting** | **capabilities** [ **module** *number* ] | **description** | **etherchannel** | **flowcontrol** | **link** [ **module** *number* ] | **private-vlan mapping** | **pruning** | **stats** | **status** [{ **err-disabled** | **inactive** }] | **trunk** }]

## Syntax Description

| | |
|---|---|
| *interface-id* | (Optional) ID of the interface. Valid interfaces include physical ports (including type, stack member for stacking-capable switches, module, and port number) and port channels.<br>The port channel range is 1 to 48. |
| **vlan** *vlan-id* | (Optional) VLAN identification. The range is 1 to 4094. |
| **accounting** | (Optional) Displays accounting information on the interface, including active protocols and input and output packets and octets.<br>**Note** The display shows only packets processed in software; hardware-switched packets do not appear. |
| **capabilities** | (Optional) Displays the capabilities of all interfaces or the specified interface, including the features and options that you can configure on the interface. Though visible in the command line help, this option is not available for VLAN IDs. |
| **module** *number* | (Optional) Displays capabilities of all interfaces on the switch or specified stack member.<br>This option is not available if you entered a specific interface ID. |
| **description** | (Optional) Displays the administrative status and description set for interfaces.<br>**Note** The output of the **show interfaces description** command displays information of all the available interfaces whether or not the corresponding network module for these interfaces are connected. These interfaces can be configured if the network module is connected. Run the **show interface status** command to see which network modules are connected. |
| **etherchannel** | (Optional) Displays interface EtherChannel information. |
| **flowcontrol** | (Optional) Displays interface flow control information. |
| **link** [**module***number*] | (Optional) Displays the up time and down time of the interface. |

| | |
|---|---|
| **private-vlan mapping** | (Optional) Displays private-VLAN mapping information for the VLAN switch virtual interfaces (SVIs). This keyword is not available if the switch is running the LAN base feature set. |
| **pruning** | (Optional) Displays trunk VTP pruning information for the interface. |
| **stats** | (Optional) Displays the input and output packets by switching the path for the interface. |
| **status** | (Optional) Displays the status of the interface. A status of unsupported in the Type field means that a non-Cisco small form-factor pluggable (SFP) module is inserted in the module slot. |
| **err-disabled** | (Optional) Displays interfaces in an error-disabled state. |
| **inactive** | (Optional) Displays interfaces in an inactive state. |
| **trunk** | (Optional) Displays interface trunk information. If you do not specify an interface, only information for active trunking ports appears. |

**Note** Though visible in the command-line help strings, the **crb**, **fair-queue**, **irb**, **mac-accounting**, **precedence**, **random-detect**, **rate-limit**, and **shape** keywords are not supported.

### Command Default

None

### Command Modes

Privileged EXEC (#)

### Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |
| Cisco IOS XE Gibraltar 16.12.1 | The **link** keyword was introduced. |

### Usage Guidelines

The **show interfaces capabilities** command with different keywords has these results:

- Use the **show interface capabilities module** *number* command to display the capabilities of all interfaces on that switch in the stack. If there is no switch with that module number in the stack, there is no output.
- Use the **show interfaces** *interface-id* **capabilities** to display the capabilities of the specified interface.
- Use the **show interfaces capabilities** (with no module number or interface ID) to display the capabilities of all interfaces in the stack.

**Note** The field **Last Input** displayed in the command output indicates the number of hours, minutes, and seconds since the last packet was successfully received by an interface and processed by the CPU on the device. This information can be used to know when a dead interface failed.

**Last Input** is not updated by fast-switched traffic.

The field **output** displayed in the command output indicates the number of hours, minutes, and seconds since the last packet was successfully transmitted by the interface. The information provided by this field can useful for knowing when a dead interface failed.

The **show interfaces link** command with different keywords has these results:

- Use the **show interface link module** *number* command to display the up time and down time of all interfaces on that switch in the stack. If there is no switch with that module number in the stack, there is no output.

**Note** On a standalone switch, the **module** *number* refers to the slot number.

- Use the **show interfaces** *interface-id* **link** to display the up time and down time of the specified interface.
- Use the **show interfaces link** (with no module number or interface ID) to display the up time and down time of all interfaces in the stack.
- If the interface is up, the up time displays the time (hours, minutes, and seconds) and the down time displays 00:00:00.
- If the interface is down, only the down time displays the time (hours, minutes, and seconds).

## Examples

This is an example of output from the **show interfaces** command for an interface on stack member 3:

```
Device# show interfaces gigabitethernet3/0/2

GigabitEthernet3/0/2 is down, line protocol is down (notconnect)
  Hardware is Gigabit Ethernet, address is 2037.064d.4381 (bia 2037.064d.4381)
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed, media type is 10/100/1000BaseTX
  input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

```
Device# show interfaces accounting

Vlan1
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                      IP          0          0          6        378
Vlan200
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
No traffic sent or received on this interface.
GigabitEthernet0/0
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                   Other     165476   11417844          0          0
           Spanning Tree    1240284   64494768          0          0
                     ARP       7096     425760          0          0
                     CDP      41368   18781072      82908   35318808
GigabitEthernet1/0/1
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
No traffic sent or received on this interface.
GigabitEthernet1/0/2
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
No traffic sent or received on this interface.

<output truncated>
```

This is an example of output from the **show interfaces** *interface* **description** command when the interface has been described as *Connects to Marketing* by using the **description** interface configuration command:

```
Device# show interfaces gigabitethernet1/0/2 description

Interface                      Status         Protocol Description
Gi1/0/2                        up             down     Connects to Marketing
```

```
Device# show interfaces etherchannel
----
Port-channel34:
Age of the Port-channel   = 28d:18h:51m:46s
Logical slot/port   = 12/34          Number of ports = 0
GC                  = 0x00000000      HotStandBy port = null
Passive port list   =
Port state          = Port-channel L3-Ag Ag-Not-Inuse
Protocol            =    -
Port security       = Disabled
```

This is an example of output from the **show interfaces** *interface-id* **pruning** command when pruning is enabled in the VTP domain:

```
Device# show interfaces gigabitethernet1/0/2 pruning

Port      Vlans pruned for lack of request by neighbor
Gi1/0/2   3,4

Port      Vlans traffic requested of neighbor
```

```
Gi1/0/2   1-3
```

This is an example of output from the **show interfaces stats** command for a specified VLAN interface:

```
Device# show interfaces vlan 1 stats

Switching path    Pkts In    Chars In    Pkts Out    Chars Out
     Processor    1165354   136205310      570800     91731594
   Route cache          0           0           0            0
         Total    1165354   136205310      570800     91731594
```

This is an example of output from the **show interfaces status err-disabled** command. It displays the status of interfaces in the error-disabled state:

```
Device# show interfaces status err-disabled

Port      Name       Status         Reason
Gi1/0/2              err-disabled   gbic-invalid
Gi2/0/3              err-disabled   dtp-flap
```

This is an example of output from the **show interfaces** *interface-id* **pruning** command:

```
Device# show interfaces gigabitethernet1/0/2 pruning

Port Vlans pruned for lack of request by neighbor


Device# show interfaces gigabitethernet1/0/1 trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi1/0/1     on               802.1q         other         10

Port        Vlans allowed on trunk
Gi1/0/1     none

Port        Vlans allowed and active in management domain
Gi1/0/1     none

Port        Vlans in spanning tree forwarding state and not pruned
Gi1/0/1     none
```

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

```
Device# show interfaces description

Interface                      Status         Protocol Description
Vl1                            admin down     down
Gi0/0                          down           down
Gi1/0/1                        down           down
Gi1/0/2                        down           down
Gi1/0/3                        down           down
Gi1/0/4                        down           down
Gi1/0/5                        down           down
Gi1/0/6                        down           down
Gi1/0/7                        down           down

<output truncated>
```

The following is a sample output of the **show interfaces link** command:

```
Device> enable
Device# show interfaces link
Port           Name                Down Time      Up Time
Gi1/0/1                            6w0d
Gi1/0/2                            6w0d
Gi1/0/3                            00:00:00       5w3d
Gi1/0/4                            6w0d
Gi1/0/5                            6w0d
Gi1/0/6                            6w0d
Gi1/0/7                            6w0d
Gi1/0/8                            6w0d
Gi1/0/9                            6w0d
Gi1/0/10                           6w0d
Gi1/0/11                           2d17h
Gi1/0/12                           6w0d
Gi1/0/13                           6w0d
Gi1/0/14                           6w0d
Gi1/0/15                           6w0d
Gi1/0/16                           6w0d
Gi1/0/17                           6w0d
Gi1/0/18                           6w0d
Gi1/0/19                           6w0d
Gi1/0/20                           6w0d
Gi1/0/21                           6w0d
```

# show interfaces counters

To display various counters for the switch or for a specific interface, use the **show interfaces counters** command in privileged EXEC mode.

**show interfaces** [*interface-id*] **counters** [{**errors** | **etherchannel** | **module** *member-number* | **protocol status** | **trunk**}]

## Syntax Description

| | |
|---|---|
| *interface-id* | (Optional) ID of the physical interface, including type, stack member (stacking-capable switches only) module, and port number. |
| **errors** | (Optional) Displays error counters. |
| **etherchannel** | (Optional) Displays EtherChannel counters, including octets, broadcast packets, multicast packets, and unicast packets received and sent. |
| **module** *member-number* | (Optional) Displays counters for the specified member.<br>The range is 1 to 9.<br>**Note** In this command, the **module** keyword refers to the stack member number. The module number that is part of the interface ID is always zero. |
| **protocol status** | (Optional) Displays the status of protocols enabled on interfaces. |
| **trunk** | (Optional) Displays trunk counters. |

**Note** Though visible in the command-line help string, the **vlan** *vlan-id* keyword is not supported.

## Command Default

None

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

If you do not enter any keywords, all counters for all interfaces are included.

This is an example of partial output from the **show interfaces counters** command. It displays all counters for the switch.

```
Device# show interfaces counters
Port            InOctets    InUcastPkts    InMcastPkts    InBcastPkts
Gi1/0/1                0              0              0              0
Gi1/0/2                0              0              0              0
Gi1/0/3         95285341          43115        1178430           1950
Gi1/0/4                0              0              0              0
```

```
<output truncated>
```

This is an example of partial output from the **show interfaces counters module** command for module 2. It displays all counters for the specified switch in the module.

```
Device# show interfaces counters module 2
Port            InOctets    InUcastPkts    InMcastPkts    InBcastPkts
Gi1/0/1              520              2              0              0
Gi1/0/2              520              2              0              0
Gi1/0/3              520              2              0              0
Gi1/0/4              520              2              0              0

<output truncated>
```

This is an example of partial output from the **show interfaces counters protocol status** command for all interfaces:

```
Device# show interfaces counters protocol status
Protocols allocated:
Vlan1: Other, IP
Vlan20: Other, IP, ARP
Vlan30: Other, IP, ARP
Vlan40: Other, IP, ARP
Vlan50: Other, IP, ARP
Vlan60: Other, IP, ARP
Vlan70: Other, IP, ARP
Vlan80: Other, IP, ARP
Vlan90: Other, IP, ARP
Vlan900: Other, IP, ARP
Vlan3000: Other, IP
Vlan3500: Other, IP
GigabitEthernet1/0/1: Other, IP, ARP, CDP
GigabitEthernet1/0/2: Other, IP
GigabitEthernet1/0/3: Other, IP
GigabitEthernet1/0/4: Other, IP
GigabitEthernet1/0/5: Other, IP
GigabitEthernet1/0/6: Other, IP
GigabitEthernet1/0/7: Other, IP
GigabitEthernet1/0/8: Other, IP
GigabitEthernet1/0/9: Other, IP
GigabitEthernet1/0/10: Other, IP, CDP

<output truncated>
```

This is an example of output from the **show interfaces counters trunk** command. It displays trunk counters for all interfaces.

```
Device# show interfaces counters trunk
Port        TrunkFramesTx   TrunkFramesRx   WrongEncap
Gi1/0/1                 0               0            0
Gi1/0/2                 0               0            0
Gi1/0/3             80678               0            0
Gi1/0/4             82320               0            0
Gi1/0/5                 0               0            0

<output truncated>
```

# show interfaces switchport

To display the administrative and operational status of a switching (nonrouting) port, including port blocking and port protection settings, use the **show interfaces switchport** command in privileged EXEC mode.

**show interfaces** [*interface-id*] **switchport** [{**module** *number*}]

## Syntax Description

| | |
|---|---|
| *interface-id* | (Optional) ID of the interface. Valid interfaces include physical ports (including type, stack member for stacking-capable switches, module, and port number) and port channels. The port channel range is 1 to 48. |
| **module** *number* | (Optional) Displays switchport configuration of all interfaces on the switch or specified stack member.<br>This option is not available if you entered a specific interface ID. |

## Command Default

None

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use the **show interface switchport module** *number* command to display the switch port characteristics of all interfaces on that switch in the stack. If there is no switch with that module number in the stack, there is no output.

This is an example of output from the **show interfaces switchport** command for a port. The table that follows describes the fields in the display.

```
Device# show interfaces gigabitethernet1/0/1 switchport
Name: Gi1/0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: down
Administrative Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 10 (VLAN0010)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: 11-20
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
```

| Field | Description |
|---|---|
| Name | Displays the port name. |
| Switchport | Displays the administrative and operational status of the port. In this display, the port is in switchport mode. |
| Administrative Mode<br>Operational Mode | Displays the administrative and operational modes. |
| Administrative Trunking Encapsulation<br>Operational Trunking Encapsulation<br>Negotiation of Trunking | Displays the administrative and operational encapsulation method and whether trunking negotiation is enabled. |
| Access Mode VLAN | Displays the VLAN ID to which the port is configured. |
| Trunking Native Mode VLAN<br>Trunking VLANs Enabled<br>Trunking VLANs Active | Lists the VLAN ID of the trunk that is in native mode. Lists the allowed VLANs on the trunk. Lists the active VLANs on the trunk. |
| Pruning VLANs Enabled | Lists the VLANs that are pruning-eligible. |
| Protected | Displays whether or not protected port is enabled (True) or disabled (False) on the interface. |
| Unknown unicast blocked<br>Unknown multicast blocked | Displays whether or not unknown multicast and unknown unicast traffic is blocked on the interface. |
| Voice VLAN | Displays the VLAN ID on which voice VLAN is enabled. |
| Appliance trust | Displays the class of service (CoS) setting of the data packets of the IP phone. |

# show interfaces transceiver

To display the physical properties of a small form-factor pluggable (SFP) module interface, use the **show interfaces transceiver** command in EXEC mode.

**show interfaces** [*interface-id*] **transceiver** [{**detail** | **module** *number* | **properties** | **supported-list** | **threshold-table**}]

## Syntax Description

| | |
|---|---|
| *interface-id* | (Optional) ID of the physical interface, including type, stack member (stacking-capable switches only) module, and port number. |
| **detail** | (Optional) Displays calibration properties, including high and low numbers and any alarm information for any Digital Optical Monitoring (DoM)-capable transceiver if one is installed in the switch. |
| **module** *number* | (Optional) Limits display to interfaces on module on the switch.<br>This option is not available if you entered a specific interface ID. |
| **properties** | (Optional) Displays speed, duplex, and inline power settings on an interface. |
| **supported-list** | (Optional) Lists all supported transceivers. |
| **threshold-table** | (Optional) Displays alarm and warning threshold table. |

## Command Modes

User EXEC (>)

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Examples

This is an example of output from the **show interfaces** *interface-id* **transceiver properties** command:

```
Device# show interfaces transceiver

If device is externally calibrated, only calibrated values are printed.
++ : high alarm, +  : high warning, -  : low warning, -- : low alarm.
NA or N/A: not applicable, Tx: transmit, Rx: receive.
mA: milliamperes, dBm: decibels (milliwatts).

                                            Optical   Optical
           Temperature  Voltage  Current    Tx Power  Rx Power
Port       (Celsius)    (Volts)  (mA)       (dBm)     (dBm)
---------  -----------  -------  --------   --------  --------
Gi5/1/2      42.9       3.28      22.1       -5.4      -8.1
Te5/1/3      32.0       3.28      19.8        2.4      -4.2


Device# show interfaces gigabitethernet1/1/1 transceiver properties
  Name : Gi1/1/1
  Administrative Speed: auto
  Operational Speed: auto
  Administrative Duplex: auto
  Administrative Power Inline: enable
  Operational Duplex: auto
  Administrative Auto-MDIX: off
  Operational Auto-MDIX: off
```

This is an example of output from the **show interfaces** *interface-id* **transceiver detail** command:

```
Device# show interfaces gigabitethernet1/1/1 transceiver detail
  ITU Channel not available (Wavelength not available),
  Transceiver is internally calibrated.
  mA:milliamperes, dBm:decibels (milliwatts), N/A:not applicable.
  ++:high alarm, +:high warning, -:low warning, -- :low alarm.
  A2D readouts (if they differ), are reported in parentheses.
  The threshold values are uncalibrated.

                              High Alarm  High Warn  Low Warn   Low Alarm
           Temperature        Threshold   Threshold  Threshold  Threshold
  Port     (Celsius)         (Celsius)   (Celsius)  (Celsius)  (Celsius)
  -------  -----------------  ----------  ---------  ---------  ---------
  Gi1/1/1  29.9               74.0        70.0       0.0        -4.0
                              High Alarm  High Warn  Low Warn   Low Alarm
           Voltage            Threshold   Threshold  Threshold  Threshold
  Port     (Volts)            (Volts)     (Volts)    (Volts)    (Volts)
  -------  ---------------    ----------  ---------  ---------  ---------
  Gi1/1/1  3.28               3.60        3.50       3.10       3.00

           Optical            High Alarm  High Warn  Low Warn   Low Alarm
           Transmit Power     Threshold   Threshold  Threshold  Threshold
  Port     (dBm)              (dBm)       (dBm)      (dBm)      (dBm)
  -------  -----------------  ----------  ---------  ---------  ---------
  Gi1/1/1  1.8                7.9         3.9        0.0        -4.0

           Optical            High Alarm  High Warn  Low Warn   Low Alarm
           Receive Power      Threshold   Threshold  Threshold  Threshold
  Port     (dBm)              (dBm)       (dBm)      (dBm)      (dBm)
  -------  -----------------  ----------  ---------  ---------  ---------
  Gi1/1/1  -23.5              -5.0        -9.0       -28.2      -32.2
```

```
Device# show interfaces transceiver supported-list
Transceiver Type           Cisco p/n min version
                               supporting DOM
------------------       -------------------------

   DWDM GBIC                ALL
   DWDM SFP                 ALL
   RX only WDM GBIC         ALL
   DWDM XENPAK              ALL
   DWDM X2                  ALL
   DWDM XFP                 ALL
   CWDM GBIC                NONE
   CWDM X2                  ALL
   CWDM XFP                 ALL
   XENPAK ZR                ALL
   X2 ZR                    ALL
   XFP ZR                   ALL
   Rx_only_WDM_XENPAK       ALL
   XENPAK_ER                10-1888-04
   X2_ER                    ALL
   XFP_ER                   ALL
   XENPAK_LR                10-1838-04
   X2_LR                    ALL
   XFP_LR                   ALL
   XENPAK_LW                ALL
   X2_LW                    ALL
   XFP_LW                   NONE
   XENPAK SR                NONE
   X2 SR                    ALL
   XFP SR                   ALL
   XENPAK LX4               NONE
   X2 LX4                   NONE
   XFP LX4                  NONE
   XENPAK CX4               NONE
   X2 CX4                   NONE
   XFP CX4                  NONE
   SX GBIC                  NONE
   LX GBIC                  NONE
   ZX GBIC                  NONE
   CWDM_SFP                 ALL
   Rx_only_WDM_SFP          NONE
   SX_SFP                   ALL
   LX_SFP                   ALL
   ZX_SFP                   ALL
   EX_SFP                   ALL
   SX SFP                   NONE
   LX SFP                   NONE
   ZX SFP                   NONE
   GIgE BX U SFP            NONE
   GigE BX D SFP            ALL
   X2 LRM                   ALL
   SR_SFPP                  ALL
   LR_SFPP                  ALL
   LRM_SFPP                 ALL
   ER_SFPP                  ALL
   ZR_SFPP                  ALL
   DWDM_SFPP                ALL
   GIgE BX 40U SFP          ALL
   GigE BX 40D SFP          ALL
   GigE BX 40DA SFP         ALL
   GIgE BX 80U SFP          ALL
   GigE BX 80D SFP          ALL
   GIG BXU_SFPP             ALL
   GIG BXD_SFPP             ALL
   GIG BX40U_SFPP           ALL
   GIG BX40D_SFPP           ALL
   GigE Dual Rate LX SFP    ALL
   CWDM_SFPP                ALL
   CPAK_SR10                ALL
   CPAK_LR4                 ALL
   QSFP_LR                  ALL
   QSFP_SR                  ALL
```

This is an example of output from the **show interfaces transceiver threshold-table** command:

```
Device# show interfaces transceiver threshold-table
              Optical Tx     Optical Rx    Temp     Laser Bias    Voltage
                                                    current
             -------------  -------------  ------   ------------  ---------

 DWDM GBIC
Min1             -4.00        -32.00        -4            N/A          4.65
Min2              0.00        -28.00        0             N/A          4.75
Max2              4.00         -9.00        70            N/A          5.25
Max1              7.00         -5.00        74            N/A          5.40
 DWDM SFP
```

```
Min1             -4.00        -32.00        -4            N/A          3.00
Min2              0.00        -28.00        0             N/A          3.10
Max2              4.00         -9.00        70            N/A          3.50
Max1              8.00         -5.00        74            N/A          3.60
 RX only WDM GBIC
Min1              N/A         -32.00        -4            N/A          4.65
Min2              N/A         -28.30        0             N/A          4.75
Max2              N/A          -9.00        70            N/A          5.25
Max1              N/A          -5.00        74            N/A          5.40
 DWDM XENPAK
Min1             -5.00        -28.00        -4            N/A          N/A
Min2             -1.00        -24.00        0             N/A          N/A
Max2              3.00         -7.00        70            N/A          N/A
Max1              7.00         -3.00        74            N/A          N/A
 DWDM X2
Min1             -5.00        -28.00        -4            N/A          N/A
Min2             -1.00        -24.00        0             N/A          N/A
Max2              3.00         -7.00        70            N/A          N/A
Max1              7.00         -3.00        74            N/A          N/A
 DWDM XFP
Min1             -5.00        -28.00        -4            N/A          N/A
Min2             -1.00        -24.00        0             N/A          N/A
Max2              3.00         -7.00        70            N/A          N/A
Max1              7.00         -3.00        74            N/A          N/A
 CWDM X2
Min1              N/A          N/A          0             N/A          N/A
Min2              N/A          N/A          0             N/A          N/A
Max2              N/A          N/A          0             N/A          N/A
Max1              N/A          N/A          0             N/A          N/A

<output truncated>
```

## Related Commands

| Command | Description |
|---|---|
| **transceiver type all** | Enters the transceiver type configuration mode. |
| **monitoring** | Enables digital optical monitoring. |

# show macro auto

To display Auto Smartports macro information, use the **show macro auto** command in user EXEC mode.

**show macro auto** {**address-group** *address-group-name* | **device** [**access-point**] [**ip-camera**] [**lightweight-ap**] [**media-player**] [**phone**] [**router**] [**switch**] | **global** [*event_trigger*] | **interface** [*interface_id*]}

## Syntax Description

| | |
|---|---|
| **address-group** [*address-group-name*] | Displays address-group information.<br>(Optional) *address-group-name*—Displays information for the specified address group. |
| **device** [*access-point*] [*ip-camera*] [*lightweight-ap*] [*media-player*] [*phone*] [*router*] [*switch*] | Displays device information about one or more devices.<br>• (Optional) **access-point**—Autonomous access point<br>• (Optional) **ip-camera**—Cisco IP video surveillance camera<br>• (Optional) **lightweight-ap**—Lightweight access point<br>• (Optional) **media-player**—Digital media player<br>• (Optional) phone—Cisco IP phone<br>• (Optional) **router**—Cisco router<br>• (Optional) **switch**—Cisco switch |
| **global** [*event_trigger*] | Displays Auto Smartports information about the switch.<br>(Optional) *event_trigger*—Displays information about the specified event trigger. |
| **interface** [*interface_id*] | Displays interface status.<br>(Optional) *interface_id*—isplays information about the specified interface. |

## Command Modes

User EXEC (>)

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

Use this command to display the Auto SmartPorts information for the switch. Use the **show macro auto device** command to display the configurable parameters for a device.

### Example

This example shows how to use the **show macro auto device** to view the configuration on the switch:

```
Device# show macro auto device
Device:lightweight-ap
Default Macro:CISCO_LWAP_AUTO_SMARTPORT
Current Macro:CISCO_LWAP_AUTO_SMARTPORT
Configurable Parameters:ACCESS_VLAN
Defaults Parameters:ACCESS_VLAN=1
Current Parameters:ACCESS_VLAN=1

Device:access-point
Default Macro:CISCO_AP_AUTO_SMARTPORT
Current Macro:CISCO_AP_AUTO_SMARTPORT
Configurable Parameters:NATIVE_VLAN
Defaults Parameters:NATIVE_VLAN=1
Current Parameters:NATIVE_VLAN=1

Device:phone
Default Macro:CISCO_PHONE_AUTO_SMARTPORT
Current Macro:CISCO_PHONE_AUTO_SMARTPORT
Configurable Parameters:ACCESS_VLAN VOICE_VLAN
Defaults Parameters:ACCESS_VLAN=1 VOICE_VLAN=2
Current Parameters:ACCESS_VLAN=1 VOICE_VLAN=2

Device:router
Default Macro:CISCO_ROUTER_AUTO_SMARTPORT
Current Macro:CISCO_ROUTER_AUTO_SMARTPORT
Configurable Parameters:NATIVE_VLAN
Defaults Parameters:NATIVE_VLAN=1
Current Parameters:NATIVE_VLAN=1

Device:switch
Default Macro:CISCO_SWITCH_AUTO_SMARTPORT
Current Macro:CISCO_SWITCH_AUTO_SMARTPORT
Configurable Parameters:NATIVE_VLAN
Defaults Parameters:NATIVE_VLAN=1
Current Parameters:NATIVE_VLAN=1

Device:ip-camera
Default Macro:CISCO_IP_CAMERA_AUTO_SMARTPORT
Current Macro:CISCO_IP_CAMERA_AUTO_SMARTPORT
Configurable Parameters:ACCESS_VLAN
Defaults Parameters:ACCESS_VLAN=1
Current Parameters:ACCESS_VLAN=1

Device:media-player
Default Macro:CISCO_DMP_AUTO_SMARTPORT
Current Macro:CISCO_DMP_AUTO_SMARTPORT
Configurable Parameters:ACCESS_VLAN
Defaults Parameters:ACCESS_VLAN=1
Current Parameters:ACCESS_VLAN=1
```

This example shows how to use the **show macro auto address-group name** command to view the TEST3 address group configuration on the switch:

```
Device# show macro auto address-group TEST3MAC Address Group Configuration:
```

```
Group Name OUI  MAC ADDRESS
--------------------------------------------------------------
TEST3 2233.33     0022.0022.0022
2233.34
```

# show memory platform

To display memory statistics of a platform, use the **show memory platform** command in privileged EXEC mode.

**show memory platform** [{**compressed-swap** | **information** | **page-merging**}]

## Syntax Description

| | |
|---|---|
| **compressed-swap** | (Optional) Displays platform memory compressed-swap information. |
| **information** | (Optional) Displays general information about the platform. |
| **page-merging** | (Optional) Displays platform memory page-merging information. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Free memory is accurately computed and displayed in the Free Memory field of the command output.

## Examples

The following is sample output from the **show memory platform** command:

```
Switch# show memory platform

 Virtual memory   : 12874653696
  Pages resident   : 627041
  Major page faults: 2220
  Minor page faults: 2348631

  Architecture     : mips64
  Memory (kB)
    Physical       : 3976852
    Total          : 3976852
    Used           : 2761276
    Free           : 1215576
    Active         : 2128196
    Inactive       : 1581856
    Inact-dirty    : 0
    Inact-clean    : 0
    Dirty          : 0
    AnonPages      : 1294984
    Bounce         : 0
    Cached         : 1978168
    Commit Limit   : 1988424
    Committed As   : 3343324
    High Total     : 0
    High Free      : 0
    Low Total      : 3976852
    Low Free       : 1215576
    Mapped         : 516316
    NFS Unstable   : 0
    Page Tables    : 17124
    Slab           : 0
    VMmalloc Chunk : 1069542588
    VMmalloc Total : 1069547512
    VMmalloc Used  : 2588
    Writeback      : 0
    HugePages Total: 0
    HugePages Free : 0
    HugePages Rsvd : 0
    HugePage Size  : 2048

  Swap (kB)
    Total          : 0
    Used           : 0
    Free           : 0
    Cached         : 0

  Buffers (kB)     : 437136

  Load Average
    1-Min          : 1.04
    5-Min          : 1.16
    15-Min         : 0.94
```

The following is sample output from the **show memory platform information** command:

```
Device# show memory platform information

 Virtual memory   : 12870438912
  Pages resident   : 626833
  Major page faults: 2222
  Minor page faults: 2362455

  Architecture     : mips64
  Memory (kB)
    Physical       : 3976852
    Total          : 3976852
    Used           : 2761224
    Free           : 1215628
    Active         : 2128060
    Inactive       : 1584444
    Inact-dirty    : 0
    Inact-clean    : 0
    Dirty          : 284
    AnonPages      : 1294656
    Bounce         : 0
    Cached         : 1979644
    Commit Limit   : 1988424
    Committed As   : 3342184
    High Total     : 0
    High Free      : 0
    Low Total      : 3976852
    Low Free       : 1215628
    Mapped         : 516212
    NFS Unstable   : 0
    Page Tables    : 17096
    Slab           : 0
    VMmalloc Chunk : 1069542588
    VMmalloc Total : 1069547512
    VMmalloc Used  : 2588
    Writeback      : 0
    HugePages Total: 0
    HugePages Free : 0
    HugePages Rsvd : 0
    HugePage Size  : 2048
```

```
Swap (kB)
  Total           : 0
  Used            : 0
  Free            : 0
  Cached          : 0

Buffers (kB)      : 438228

Load Average
  1-Min           : 1.54
  5-Min           : 1.27
  15-Min          : 0.99
```

# show module

To display module information such as switch number, model number, serial number, hardware revision number, software version, MAC address and so on, use this command in user EXEC or privileged EXEC mode.

```
show module [{switch-num}]
```

**Syntax Description**

| | |
|---|---|
| *switch-num* | (Optional) Number of the switch. |

**Command Default**

None

**Command Modes**

User EXEC (>)

Privileged EXEC (#)

**Command History**

| **Release** | **Modification** |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

Entering the **show module** command without the *switch-num* argument is the same as entering the show module all command.

# show network-policy profile

To display the network-policy profiles, use the **show network policy profile** command in privileged EXEC mode.

**show network-policy profile** [*profile-number*] [**detail**]

## Syntax Description

| | |
|---|---|
| *profile-number* | (Optional) Displays the network-policy profile number. If no profile is entered, all network-policy profiles appear. |
| **detail** | (Optional) Displays detailed status and statistics information. |

## Command Default

None

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

This is an example of output from the **show network-policy profile** command:

```
Device# show network-policy profile
Network Policy Profile 10
   voice vlan 17 cos 4
  Interface:
   none
Network Policy Profile 30
   voice vlan 30 cos 5
  Interface:
   none
Network Policy Profile 36
   voice vlan 4 cos 3
  Interface:
   Interface_id
```

# show parser macro

To display the parameters for all configured macros or for one macro on the switch, use the **show parser macro** command in user EXEC mode.

**show parser macro** {**brief** | **description** [**interface** *interface-id*] | **name** *macro-name*}

**Syntax Description**

| | |
|---|---|
| **brief** | (Optional) Displays the name of each macro. |
| **description** [**interface** *interface-id*] | (Optional) Displays all macro descriptions or the description of a specific interface. |
| **name** *macro-name* | (Optional) Displays information about a single macro identified by the macro name. |

**Command Modes**

User EXEC (>)

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

### Example

This is a partial output example from the **show parser macro** command. The output for the Cisco-default macros varies depending on the switch platform and the software image running on the switch:

```
Device# show parser macro
Total number of macros = 6
--------------------------------------------------------------
Macro name : cisco-global
Macro type : default global
# Enable dynamic port error recovery for link state
# failures
errdisable recovery cause link-flap
errdisable recovery interval 60

<output truncated>

--------------------------------------------------------------
Macro name : cisco-desktop
Macro type : default interface
# macro keywords $AVID
# Basic interface - Enable data VLAN only
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID
switchport mode access

<output truncated>

--------------------------------------------------------------
Macro name : cisco-phone
Macro type : default interface
# Cisco IP phone + desktop template
# macro keywords $AVID $VVID
# VoIP enabled interface - Enable data VLAN
# and voice VLAN (VVID)
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID
switchport mode access

<output truncated>

--------------------------------------------------------------
Macro name : cisco-switch
Macro type : default interface
# macro keywords $NVID
# Access Uplink to Distribution
# Do not apply to EtherChannel/Port Group
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID

<output truncated>

--------------------------------------------------------------
Macro name : cisco-router
Macro type : default interface
# macro keywords $NVID
# Access Uplink to Distribution
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID

<output truncated>

--------------------------------------------------------------
Macro name : snmp
Macro type : customizable

#enable port security, linkup, and linkdown traps
snmp-server enable traps port-security
snmp-server enable traps linkup
snmp-server enable traps linkdown
#set snmp-server host
snmp-server host ADDRESS
#set SNMP trap notifications precedence
snmp-server ip precedence VALUE

--------------------------------------------------------------
```

This example shows the output from the **show parser macro name** command:

```
Device# show parser macro name standard-switch10
Macro name : standard-switch10
Macro type : customizable
macro description standard-switch10
# Trust QoS settings on VOIP packets
auto qos voip trust
# Allow port channels to be automatically formed
channel-protocol pagp
```

This example shows the output from the **show parser macro brief** command:

```
Device# show parser macro brief
    default global   : cisco-global
    default interface: cisco-desktop
    default interface: cisco-phone
    default interface: cisco-switch
    default interface: cisco-router
    customizable     : snmp
```

This exampe shows the output from the **show parser macro description** command:

```
Device# show parser macro description
Global Macro(s): cisco-global
Interface    Macro Description(s)
--------------------------------------------------------------
Gi1/0/1        standard-switch10
Gi1/0/2       this is test macro
--------------------------------------------------------------
```

This example shows the output from the **show parser macro description interface** command:

```
Device# show parser macro description interface gigabitethernet1/0/2
Interface    Macro Description
--------------------------------------------------------------
Gi1/0/2      this is test macro
--------------------------------------------------------------
```

# show platform hardware bluetooth

To display information about Bluetooth interface, use the **show platform hardware bluetooth** command in privileged EXEC mode.

**show platform hardware bluetooth**

**Command Default** None

**Command Modes** Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Gibraltar 16.12.1 | This command was introduced. |

**Usage Guidelines** The **show platform hardware bluetooth** command is to be used when an external USB Bluetooth dongle is connected on the device.

**Examples**

This example shows how to display the information of the Bluetooth interface using the **show platform hardware bluetooth** command.

```
Device> enable
Device# show platform hardware bluetooth
Controller: 0:1a:7d:da:71:13
Type: Primary
Bus: USB
State: DOWN
Name:
HCI Version:
```

# show platform hardware fed switch forward interface

To debug forwarding information and to trace the packet path in the hardware forwarding plane, use the **show platform hardware fed switch** *switch_number* **forward interface** command. This command simulates a user-defined packet and retrieves the forwarding information from the hardware forwarding plane. A packet is generated on the ingress port based on the packet parameters that you have specified in this command. You can also provide a complete packet from the captured packets stored in a PCAP file.

This topic elaborates only the interface forwarding-specific options, that is, the options available with the **show platform hardware fed switch** {*switch_num* | **active** | **standby** } **forward interface** command.

**show platform hardware fed switch** {*switch_num* | **active** | **standby**} **forward interface** *interface-type interface-number* **source-mac-address** *destination-mac-address*{*protocol-number* | **arp** | **cos** | **ipv4** | **ipv6** | **mpls**}

**show platform hardware fed switch** {*switch_num* | **active** | **standby**} **forward interface** *interface-type interface-number* **pcap** *pcap-file-name* **number** *packet-number* **data**

**show platform hardware fed switch** {*switch_num* | **active** | **standby**} **forward interface** *interface-type interface-number* **vlan** *vlan-id source-mac-address destination-mac-address*{*protocol-number* | **arp** | **cos** | **ipv4** | **ipv6** | **mpls**}

## Syntax Description

| | |
|---|---|
| **switch** {*switch_num* \| **active** \| **standby** } | The switch on which packet tracing has to be scheduled. The input port should be available on this switch. You have the following options :<br>• *switch_num*—ID of the switch on which the ingress port is present.<br>• **active**—indicates the active switch on which the the ingress port is present.<br>• **standby**—indicates the standby switch on which the ingress port is present.<br>**Note** This keyword is not supported. |
| **interface** *interface-type interface-number* | The input interface on which packet trace is simulated. |
| *source-mac-address* | The source MAC address of the packet you want to simulate. |
| *destination-mac-address* | The MAC address of the destination interface in hexadecimal format. |
| *protocol-number* | The number assigned to any L3 protocol. |
| **arp** | The Address Resolution Protocol (ARP) parameters. |
| **ipv4** | The IPv4 packet parameters. |
| **ipv6** | The IPv6 packet parameters. |
| **mpls** | The Multiprotocol Label Switching (MPLS) label parameters. |

| | |
|---|---|
| **cos** | The class of service (CoS) number from 0 to 7 to set priority. |
| **pcap** *pcap-file-name* | Name of the pcap file in internal flash (flash:).<br>Ensure that the file already exists in flash:. |
| **number** *packet-number* | Specifies the packet number in the pcap file. |
| **vlan** *vlan-id* | VLAN id of the dot1q header in the simulated packet. The range is 1 to 4096. |

## Command Modes

Privileged EXEC

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |
| Cisco IOS XE Fuji 16.9.1 | The command was enhanced to support MPLS/ARP/VxLAN packet parameters and trace packets captured in a PCAP file. |
| Cisco IOS XE Gibraltar 16.10.1 | The command was enhanced to support data capture across a stack. |

## Usage Guidelines

Do not use this command unless a technical support representative asks you to. Use this command only when you are working directly with a technical support representative while troubleshooting a problem.

This command supports the following packet types:

- Non-IP packets with any L3 protocol
- ARP packets
- IPv4 packets with any L4 protocol
- IPv4 packets with TCP/UDP/IGMP/ICMP/SCTP payload
- VxLAN packets
- MPLS packets with up to 3 Labels and meta data
- MPLS packets with IPv4/IPv6 payload
- IPv6 packets with TCP/UDP/IGMP/ICMP/SCTP payload

In a stack environment, you can trace packets across the stack irrespective of the number of stack members and topology. The **show platform hardware fed switch** *switch-number* **forward interface** *interface-type interface-number* command consolidates packet-forwarding information of all the stack members on the ingress switch. To achieve this, ensure that the switch number specified in the *switch_num* and *interface-number* arguments are of the input switch and that the number matches.

To trace any particular packet from the captured packets stored in a PCAP file, use the **show platform hardware fed switch forward interface** *interface-type interface-number* **pcap** *pcap-file-name number packet-number* **data** command.

### Example

This is an example of output from the **show platform hardware fed switch** {*switch_num* | **active** | **standby** } **forward interface** command.

```
Device#show platform hardware fed switch active forward interface gigabitEthernet 1/0/35
0000.0022.0055 0000.0055.0066 ipv4 44.44.0.2 55.55.0.2 udp 1222 3333

Show forward is running in the background. After completion, syslog will be generated.

*Sep 24 05:57:36.614: %SHFWD-6-PACKET_TRACE_DONE: Switch 1 R0/0: fed: Packet Trace Complete:
  Execute (show platform hardware fed switch <> forward last summary|detail)
*Sep 24 05:57:36.614: %SHFWD-6-PACKET_TRACE_FLOW_ID: Switch 1 R0/0: fed: Packet Trace Flow
 id is 150323855361
```

### Related Commands

| Command | Description |
|---|---|
| **monitor capture interface** | Configures monitor capture points specifying an attachment point and the packet flow direction. |
| **monitor capture start** | Starts the capture of packet data at a traffic trace point into a buffer. |
| **monitor capture stop** | Stops the capture of packet data at a traffic trace point. |
| **monitor capture export** | Saves the captured packets in the buffer.<br>Use this command to export the monitor capture buffer to a pcap file in flash: that you can use as an input in the **show forward** with **pcap**. |

# show platform hardware fed switch fwd-asic counters tla

To display the register information of a counter from the forwarding ASIC, use the **show platform hardware fed switch fwd-asic counters tla** command in the Privileged EXEC mode.

**show platform hardware fed switch** {*switch_num* | **active** | **standby**} **fwd-asic counters tla** *tla_counter*{**detail** | **drop** | **statistics**} [**asic** *asic_num*] **output** *location:filename*

**Syntax Description**

| | |
|---|---|
| **switch** {*switch_num* \| **active** \| **standby** } | The switch for which you want to display information. You have the following options :<br>• *switch_num*: ID of the switch.<br>• **active**: Displays information relating to the active switch.<br>• **standby**: Displays information relating to the standby switch, if available. |

| | |
|---|---|
| **tla***tla_counter* | *tla_counter* can be any of the following Three Letter Acronym (TLA) counters:<br>• AQM Active Queue Management<br>• ASE ACL Search Engine<br>• DPP DopplerE Point to Point<br>• EGR Egress Global Resolution<br>• EPF Egress Port FIFO<br>• ESM Egress Scheduler Module<br>• EQC Egress Queue Controller<br>• FPE Flexible Parser<br>• FPS Flexible Pipe Stage<br>• FSE Fib Search Engine<br>• IGR Ingress Global Resolution<br>• IPF Ingress Port FIFO<br>• IQS Ingress Queues and Scheduler<br>• MSC Macsec Engine<br>• NFL Netflow<br>• NIF Network Interface<br>• PBC Packet Buffer Complex<br>• PIM Protocol Independent Multicast<br>• PLC Policer<br>• RMU Recirculation Multiplexer Unit<br>• RRE Reassembly Engine<br>• RWE Rewrite Engine<br>• SEC Security Engine<br>• SIF Stack Interface<br>• SPQ Supervisor Packet Queuing Engine<br>• SQS Stack Queues And Scheduler<br>• SUP Supervisor Interface |
| **detail** | Displays the contents of the registers of all non-zero counters. |
| **drop** | Displays the contents of the registers of all non-zero drop counters. |
| **statistics** | Displays the contents of the registers of all non-zero statistical counters. |

| | |
|---|---|
| **asci** *asic_num* | (Optional) Specifies the ASIC. |
| **output** *location:filename* | Specifies an output file to which the contents of the counters registers are to be dumped. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |
| Cisco IOS XE Amsterdam 17.3.1 | The command output was modified to be presented in a readable tabular format. The size of the output file was also reduced by not printing fields that had zero values.<br>The **change** keyword was deprecated. |

## Usage Guidelines

Do not use this command unless a technical support representative asks you to. Use this command only when you are working directly with a technical support representative while troubleshooting a problem.

**Note** Some TLAs may not have any registers to display as part of **drop** or **statistics** options because of the lack of these drop or statistics registers for them. In such a case, a message, `No <detail|drop|statistics> counters to display for tla <TLA_NAME>` is displayed and no output file is generated.

### Example

This is an example output from the **show platform hardware fed active fwd-asic counters tla aqm** command.

```
Device#show platform hardware fed active fwd-asic counters tla aqm detail output flash:aqm
command to get counters for tla AQM succeeded
Device#
Device# more flash:aqm
==========================================================================================

 asic | core | Register Name             | Fields                                     | value
==========================================================================================

  0     0      AqmRepTransitUsageCnt[0][0]
                                           totalCntHighMark                    : 0x4
                                           transitWait4DoneHighMark            : 0x2
  0     1      AqmRepTransitUsageCnt[0][0]
                                           totalCntHighMark                    : 0x2
                                           transitWait4DoneHighMark            : 0x2
==========================================================================================

 asic | core | Register Name             | Fields                                     | value
==========================================================================================

  0     0      AqmGlobalHardBufCnt[0][0]
```

```
                                           highWaterMark                       : 0x3
==========================================================================================

 asic | core | Register Name            | Fields                              | value
==========================================================================================

  0      0      AqmRedQueueStats[0][673]
                                           acceptByteCnt2                      : 0x4e44e
                                           acceptFrameCnt2                     : 0x5e1
  0      0      AqmRedQueueStats[0][674]
                                           acceptByteCnt1                      : 0x88
                                           acceptByteCnt2                      : 0xa7c
                                           acceptFrameCnt1                     : 0x2
                                           acceptFrameCnt2                     : 0x16
  0      0      AqmRedQueueStats[0][676]
                                           acceptByteCnt2                      : 0xfbf06
                                           acceptFrameCnt2                     : 0x2440
  0      0      AqmRedQueueStats[0][677]
                                           acceptByteCnt2                      : 0xcc
                                           acceptFrameCnt2                     : 0x3
  0      0      AqmRedQueueStats[0][687]
                                           acceptByteCnt2                      : 0x2caea0
                                           acceptFrameCnt2                     : 0xa836
  0      0      AqmRedQueueStats[0][691]
                                           acceptByteCnt2                      : 0x2dc
                                           acceptFrameCnt2                     : 0x6
  0      0      AqmRedQueueStats[0][692]
                                           acceptByteCnt2                      : 0xc518
                                           acceptFrameCnt2                     : 0x2e6
```

# show platform hardware fed active fwd-asic resource tcam utilization

To display hardware information about the Ternary Content Addressable Memory (TCAM) usage, use the **show platform hardware fed active fwd-asic resource tcam utilization** command in privileged EXEC mode.

**show platform hardware fed active fwd-asic resource tcam utilization**[{*asic-number* }]

## Syntax Description

| | |
|---|---|
| *asic-number* | ASIC number. Valid values are from 0 to 7. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Amsterdam 17.2.1 | This command was introduced in a release prior to Cisco IOS XE Amsterdam 17.2.1 . |

## Usage Guidelines

On stackable switches, this command has the **switch** keyword, **show platform hardware fed switch active fwd-asic resource tcam utilization**. On non-stackable switches, the **switch** keyword is not available.

### Example

The following is sample output from the **show platform hardware fed active fwd-asic resource tcam utilization** command:

```
Device# show platform hardware fed active fwd-asic resource tcam utilization

Codes: EM - Exact_Match, I - Input, O - Output, IO - Input & Output, NA - Not Applicable

CAM Utilization for ASIC  [0]
 Table                  Subtype      Dir      Max     Used    %Used       V4       V6
MPLS    Other

-----------------------------------------------------------------------------------------------------

 OPENFLOW Table0        TCAM         I        5000        5       0%        3        0
    0        2
 OPENFLOW Table0 Ext.   EM           I        8192        3       0%        0        0
    0        3
 OPENFLOW Table1        TCAM         I        3600        1       0%        1        0
    0        0
 OPENFLOW Table1 Ext.   EM           I        8192        1       0%        0        0
    0        1
 OPENFLOW Table2        TCAM         I        3500        1       0%        1        0
    0        0
 OPENFLOW Table2 Ext.   EM           I        8192        1       0%        0        0
    0        1
 OPENFLOW Table3 Ext.   EM           I        8192        0       0%        0        0
    0        0
 OPENFLOW Table4 Ext.   EM           I        8192        0       0%        0        0
    0        0
 OPENFLOW Table5 Ext.   EM           I        8192        0       0%         0         0
    0        0
 OPENFLOW Table6 Ext.   EM           I        8192        0       0%         0         0
    0        0
 OPENFLOW Table7 Ext.   EM           I        8192        0       0%         0         0
    0        0
```

The table below lists the significant fields shown in the display.

*Table 6: show platform hardware fed active fwd-asic resource tcam utilization Field Descriptions*

| Field | Description |
|---|---|
| Table | OpenFlow table numbers. |
| Subtype | What are the different subtypes available? |
| Dir | |
| Max | |
| Used | |
| %Used | |
| V4 | |
| V6 | |
| MPLS | |
| Other | |

# show platform resources

To display platform resource information, use the **show platform reources** command in privileged EXEC mode.

**show platform resources**

This command has no arguments or keywords.

**Command Modes**

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

The output of this command displays the used memory, which is total memory minus the accurate free memory.

## Example

The following is sample output from the **show platform resources** command:

```
Switch# show platform resources

**State Acronym: H - Healthy, W - Warning, C - Critical

Resource                 Usage                 Max             Warning         Critical
     State
-----------------------------------------------------------------------------------------

 Control Processor       7.20%                 100%            90%             95%
     H
  DRAM                   2701MB(69%)           3883MB          90%             95%
     H
```

# show platform software audit

To display the SE Linux Audit logs, use the **show platform software audit** command in privileged EXEC mode.

**show platform software audit** {**all** | **summary** | [**switch** {*switch-number* | **active** | **standby**}] {**0** | **F0** | **R0** | {**FP** | **RP**} {**active**}}}

## Syntax Description

| | |
|---|---|
| **all** | Shows the audit log from all the slots. |
| **summary** | Shows the audit log summary count from all the slots. |
| **switch** | Shows the audit logs for a slot on a specific switch. |
| *switch-number* | Selects the switch with the specified switch number. |
| **switch active** | Selects the active instance of the switch. |
| **standby** | Selects the standby instance of the switch. |
| **0** | Shows the audit log for the SPA-Inter-Processor slot 0. |
| **F0** | Shows the audit log for the Embedded-Service-Processor slot 0. |
| **R0** | Shows the audit log for the Route-Processor slot 0. |
| **FP active** | Shows the audit log for the active Embedded-Service-Processor slot. |
| **RP active** | Shows the audit log for the active Route-Processor slot. |

## Command Modes

Privileged EXEC (#)

## Command History

## Usage Guidelines

This command was introduced in the Cisco IOS XE Gibraltar 16.10.1 as a part of the SELinux Permissive Mode feature. The **show platform software audit** command displays the system logs containing the access violation events.

In Cisco IOS XE Gibraltar 16.10.1, operation in a permissive mode is available - with the intent of confining specific components (process or application) of the IOS-XE platform. In the permissive mode, access violation events are detected and system logs are generated, but the event or operation itself is not blocked. The solution operates mainly in an access violation detection mode.

The following is a sample output of the **show software platform software audit summary** command:

```
Device# show platform software audit summary

===================================
AUDIT LOG ON switch 1
-----------------------------------
```

```
AVC Denial count: 58
===================================
```

The following is a sample output of the **show software platform software audit all** command:

```
Device# show platform software audit all

===================================
AUDIT LOG ON switch 1
-----------------------------------
========== START ============
type=AVC msg=audit(1539222292.584:100): avc:  denied  { read } for  pid=14017
comm="mcp_trace_filte" name="crashinfo" dev="rootfs" ino=13667
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=lnk_file permissive=1
type=AVC msg=audit(1539222292.584:100): avc:  denied  { getattr } for  pid=14017
comm="mcp_trace_filte" path="/mnt/sd1" dev="sda1" ino=2
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:101): avc:  denied  { getattr } for  pid=14028 comm="ls"
 path="/tmp/ufs/crashinfo" dev="tmpfs" ino=58407
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:102): avc:  denied  { read } for  pid=14028 comm="ls"
name="crashinfo" dev="tmpfs" ino=58407 scontext=system_u:system_r:polaris_trace_filter_t:s0
 tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539438600.896:119): avc:  denied  { execute } for  pid=8300 comm="sh"
name="id" dev="loop0" ino=6982 scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
 tcontext=system_u:object_r:bin_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438600.897:120): avc:  denied  { execute_no_trans } for  pid=8300
comm="sh"
path="/tmp/sw/mount/cat9k-rpbase.2018-10-02_00.13_mhungund.SSA.pkg/nyquist/usr/bin/id"
dev="loop0" ino=6982 scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:bin_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438615.535:121): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
type=AVC msg=audit(1539438624.916:122): avc:  denied  { execute_no_trans } for  pid=8600
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438648.936:123): avc:  denied  { execute_no_trans } for  pid=9307
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438678.649:124): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
type=AVC msg=audit(1539438696.969:125): avc:  denied  { execute_no_trans } for  pid=10057
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438732.973:126): avc:  denied  { execute_no_trans } for  pid=10858
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438778.008:127): avc:  denied  { execute_no_trans } for  pid=11579
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438800.156:128): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
type=AVC msg=audit(1539438834.099:129): avc:  denied  { execute_no_trans } for  pid=12451
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
```

```
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539440246.697:149): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
type=AVC msg=audit(1539440299.119:150): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
========== END ============
===================================
```

The following is a sample output of the **show software platform software audit switch** command:

```
Device# show platform software audit switch active R0

========== START ============
type=AVC msg=audit(1539222292.584:100): avc:  denied  { read } for  pid=14017
comm="mcp_trace_filte" name="crashinfo" dev="rootfs" ino=13667
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=lnk_file permissive=1
type=AVC msg=audit(1539222292.584:100): avc:  denied  { getattr } for  pid=14017
comm="mcp_trace_filte" path="/mnt/sd1" dev="sda1" ino=2
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:101): avc:  denied  { getattr } for  pid=14028 comm="ls"
 path="/tmp/ufs/crashinfo" dev="tmpfs" ino=58407
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:102): avc:  denied  { read } for  pid=14028 comm="ls"
name="crashinfo" dev="tmpfs" ino=58407 scontext=system_u:system_r:polaris_trace_filter_t:s0
 tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539438624.916:122): avc:  denied  { execute_no_trans } for  pid=8600
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438648.936:123): avc:  denied  { execute_no_trans } for  pid=9307
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438678.649:124): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
type=AVC msg=audit(1539438696.969:125): avc:  denied  { execute_no_trans } for  pid=10057
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438732.973:126): avc:  denied  { execute_no_trans } for  pid=10858
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438778.008:127): avc:  denied  { execute_no_trans } for  pid=11579
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438800.156:128): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
type=AVC msg=audit(1539438834.099:129): avc:  denied  { execute_no_trans } for  pid=12451
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438860.907:130): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
```

```
========== END ============
===================================
```

# show platform software fed switch punt cpuq rates

To display the rate at which packets are punted, including the drops in the punted path, use the **show platform software fed switch punt cpuq rates** command in privileged EXEC mode.

**show platform software fed switch** {*switch-number* | **active** | **standby**} **punt cpuq rates**

## Syntax Description

| | |
|---|---|
| **switch**{*switch-number* \| **active** \| **standby**} | Displays information about the switch. You have the following options:<br>• *switch-number.*<br>• **active** —Displays information relating to the active switch.<br>• **standby**—Displays information relating to the standby switch, if available.<br>**Note** This keyword is not supported. |
| **punt** | Specifies the punt informtion. |
| **cpuq** | Specifies information about CPU receive queue. |
| **rates** | Specifies the rate at which the packets are punted. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Gibraltar 16.10.1 | This command was introduced. |

### Example

The following is sample output from the **show platform software fed switch active punt cpuq rates** command.

The output of this command displays the rate in packets per second at intervals of 10 seconds, 1 minute and 5 minutes.

```
Device#show platform software fed switch active punt cpuq rates

Punt Rate CPU Q Statistics


Packets per second averaged over 10 seconds, 1 min and 5 mins
=====================================================================================
 Q  |          Queue            |  Rx     |  Rx    |  Rx    |  Drop  |  Drop    |  Drop
 no |          Name             |  10s    |  1min  |  5min  |  10s   |  1min    |  5min
=====================================================================================
 0  CPU_Q_DOT1X_AUTH                 0        0        0        0        0        0
 1  CPU_Q_L2_CONTROL                 0        0        0        0        0        0
 2  CPU_Q_FORUS_TRAFFIC              336      266      320      0        0        0
 3  CPU_Q_ICMP_GEN                  0         0         0         0         0         0
 4  CPU_Q_ROUTING_CONTROL           0         0         0         0         0         0
 5  CPU_Q_FORUS_ADDR_RESOLUTION     0         0         0         0         0         0
 6  CPU_Q_ICMP_REDIRECT             0         0         0         0         0         0
 7  CPU_Q_INTER_FED_TRAFFIC         0         0         0         0         0         0
 8  CPU_Q_L2LVX_CONTROL_PKT         0         0         0         0         0         0
 9  CPU_Q_EWLC_CONTROL              0         0         0         0         0         0
 10 CPU_Q_EWLC_DATA                 0         0         0         0         0         0
 11 CPU_Q_L2LVX_DATA_PKT            0         0         0         0         0         0
 12 CPU_Q_BROADCAST                 0         0         0         0         0         0
 13 CPU_Q_LEARNING_CACHE_OVFL       0         0         0         0         0         0
 14 CPU_Q_SW_FORWARDING             0         0         0         0         0         0
 15 CPU_Q_TOPOLOGY_CONTROL          0         0         0         0         0         0
 16 CPU_Q_PROTO_SNOOPING            0         0         0         0         0         0
 17 CPU_Q_DHCP_SNOOPING             0         0         0         0         0         0
 18 CPU_Q_TRANSIT_TRAFFIC           0         0         0         0         0         0
 19 CPU_Q_RPF_FAILED                0         0         0         0         0         0
 20 CPU_Q_MCAST_END_STATION_SERVICE 0         0         0         0         0         0
 21 CPU_Q_LOGGING                   0         0         0         0         0         0
 22 CPU_Q_PUNT_WEBAUTH              0         0         0         0         0         0
 23 CPU_Q_HIGH_RATE_APP             0         0         0         0         0         0
 24 CPU_Q_EXCEPTION                 0         0         0         0         0         0
 25 CPU_Q_SYSTEM_CRITICAL           0         0         0         0         0         0
 26 CPU_Q_NFL_SAMPLED_DATA          0         0         0         0         0         0
 27 CPU_Q_LOW_LATENCY               0         0         0         0         0         0
 28 CPU_Q_EGR_EXCEPTION             0         0         0         0         0         0
 29 CPU_Q_FSS                       0         0         0         0         0         0
 30 CPU_Q_MCAST_DATA                0         0         0         0         0         0
 31 CPU_Q_GOLD_PKT                  0         0         0         0         0         0

------------------------------------------------------------------------------------
```

The table below describes the significant fields shown in the display.

*Table 7: **show platform software fed switch active punt cpuq rates** Field Descriptions*

| Field | Description |
|---|---|
| Queue Name | Name of the queue. |
| Rx | The rate at which the packets are received per second in 10s, 1 minute and 5 minutes. |
| Drop | The rate at which the packets are dropped per second in 10s, 1 minute and 5 minutes. |

# show platform software fed switch punt packet-capture display

To display packet capture information during high CPU utilization, use the **show platform software fed switch active punt packet-capture display** command in privileged EXEC mode.

**show platform software fed switch active punt packet-capture display** { **detailed** | **hexdump**}

## Syntax Description

| | |
|---|---|
| **switch**{*switch-number* \| **active** \| **standby**} | Displays information about a switch. You have the following options:<br>• **active** —Displays information relating to the active switch.<br>• **standby**—Displays information relating to the standby switch, if available.<br>**Note** The **standby** keyword is not supported. |
| **punt** | Specifies punt information. |
| **packet-capture display** | Specifies information about the captured packet. |
| **detailed** | Specifies detailed information about the captured packet. |
| **hex-dump** | Specifies information about the captured packet, in hex format. |

## Command Modes

Privileged EXEC (#)

## Command History

| **Release** | **Modification** |
|---|---|
| Cisco IOS XE Gibraltar 16.10.1 | This command was introduced. |

## Usage Guidelines

The output of this command displays the periodic and persistent logs of CPU-bound packets, inband CPU traffic rates, and running CPU processes when the CPU passes a high CPU utilization threshold.

## Examples

The following is a sample output from the **show platform software fed switch active punt packet-capture display detailed** command:

```
Device# show platform software fed switch active punt packet-capture display detailed
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 101 packets. Capture capacity : 4096 packets

------ Packet Number: 1, Timestamp: 2018/09/04 23:22:10.179 ------
 interface : GigabitEthernet2/0/2 [if-id: 0x00000032] (physical)
 ether hdr : dest mac: 0100.0ccc.cccd, src mac: 2c36.f8fc.4884
 ether hdr : ethertype: 0x0032

 Doppler Frame Descriptor :
     0000000044004E04  C00F402D94510000  0000000000000100  0000400401000000
     0000000001000050  000000006D000100  0000000025836200  0000000000000000

   Packet Data Dump (length: 68 bytes) :
     01000CCCCCCD2C36  F8FC48840032AAAA  0300000C010B0000  00000080012C36F8
     FC48800000000080  012C36F8FC488080  040000140002000F  0071000000020001
     244E733E

 ------ Packet Number: 2, Timestamp: 2018/09/04 23:22:10.179 ------
  interface : GigabitEthernet2/0/2 [if-id: 0x00000032] (physical)
  ether hdr : dest mac: 0180.c200.0000, src mac: 2c36.f8fc.4884
  ether hdr : ethertype: 0x0026
 !
 !
 !
```

# show platform software fed switch punt packet-capture cpu-top-talker

To display the occurrences of an attribute of a packet capture, use the **show platform softwarefed switch punt packet-capture cpu-top-talker** command in privileged EXEC mode.

**show platform software fed switch** { *switch number* | **active** | **standby** } **punt packet-capture cpu-top-talker** { **cause-code** | **dst_ipv4** | **dst_ipv6** | **dst_l4** | **dst_mac** | **eth_type** | **incoming-interface** | **ipv6_hoplt** | **protocol** | **src_dst_port** | **src_ipv4** | **src_ipv6** | **src_l4** | **src_mac** | **summary** | **ttl** | **vlan** }

**Syntax Description**

| | |
|---|---|
| **switch**{*switch-number* \| **active** \| **standby**} | Displays information about a switch. You have the following options:<br>• **active** —Displays information relating to the active switch.<br>• **standby**—Displays information relating to the standby switch, if available.<br>**Note** The **standby**keyword is not supported.<br>**Note** The **switch**keyword is not supported on nonstackable devices and on the devices that do not support StackWise Virtual. |
| **cause-code** | Displays the occurrences of cause-code. |
| **dst_ipv4** | Displays the occurrences on the destination IPv4 interface. |
| **dst_ipv6** | Displays the occurrences on the destination IPv6 interface. |
| **dst_l4** | Displays the occurrences of the Layer 4 destination port. |
| **dst_mac** | Displays the occurrences of the destination MAC address. |
| **eth_type** | Displays the occurrences of the Ethernet frame type. |
| **incoming-interface** | Displays the occurrences of incoming-interfaces. |
| **ipv6_hoplt** | Displays the occurrences of the hop limit on IPv6. |
| **protocol** | Displays the occurrences of the Layer 4 protocol. |
| **src_dst_port** | Displays the occurrences of the Layer 4 source destination port. |
| **src_ipv4** | Displays the occurrences on the source IPv4 interface. |
| **src_ipv6** | Displays the occurrences on the source IPv6 interface. |
| **src_l4** | Displays the occurrences on the Layer 4 source. |

| | |
|---|---|
| **src_mac** | Displays the occurrences of the source MAC address. |
| **summary** | Displays the summary of the occurrences of all the attributes. |
| **ttl** | Displays the occurrences on IPv4 Time to Live (TTL). |
| **vlan** | Displays the occurrences of VLAN. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Bengaluru 17.6.1 | This command was introduced. |

## Usage Guidelines

Ensure to start and stop debugging of the packets from the active switch to obtain the occurrences of the packet capture attributes.

## Examples

The following is a sample out of the **debugplatform software fed switch active punt packet-capture start**command:

```
Device# debug platform software fed active punt packet-capture start
Punt packet capturing started.
Device#
*Jan 28 12:51:14.978: %FED_PUNJECT-6-PKT_CAPTURE_FULL: F0/0: fed: Punject pkt capture buffer
 is full. Use show command to display the punted packets
```

The following is a sample out of the **debugplatform software fed switch active punt packet-capture stop**command:

```
Device# debug platform software fed active punt packet-capture stop

Punt packet capturing stopped. Captured 4096 packet(s)
```

These commands provide a maximum of ten unique values in descending order for each of the attributes.

The following is a sample output of the **show platform software fed switch active punt packet-capture cpu-top-talkercause-code** command:

```
Device# show platform software fed switch active punt packet-capture cpu-top-talker cause-code

Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets
Sr.no.    Value/Key Occurrence
1    Layer2 control protocols 4096
```

The following is a sample output of the **show platform software fed switch active punt packet-capture cpu-top-talkerdst_mac** command:

```
Device# show platform software fed switch active punt packet-capture cpu-top-talker dst_mac
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets
Sr.no.    Value/Key Occurrence
1    01:80:c2:00:00:00   4096
```

The following is a sample output of the **show platform software fed switch active punt packet-capture cpu-top-talkerincoming-interface** command:

```
Device# show platform software fed switch active punt packet-capture cpu-top-talker
incoming-interface
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets
Sr.no.    Value/Key Occurrence
1    TwentyFiveGigE1/0/1 1366
2    TwentyFiveGigE1/0/16     1365
3    TwentyFiveGigE1/0/18     1365
```

The following is a sample output of the **show platform software fed switch activepunt packet-capture cpu-top-talkersrc_mac** command:

```
Device# show platform software fed switch active punt packet-capture cpu-top-talker src_mac
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets
Sr.no.    Value/Key Occurrence
1    70:b3:17:1e:9e:8f   1366
2    70:b3:17:1e:9e:90   1365
3    70:b3:17:1e:9e:91   1365
```

The following is a sample output of the **show platform software fed switch activepunt packet-capture cpu-top-talkersummary** command. This command will provide one highest output for each of the attributes.

```
Device# show platform software fed switch active punt packet-capture cpu-top-talker summary
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets

L2 Top Talkers:
1366 Source mac     70:b3:17:1e:9e:8f
4096 Dest mac  01:80:c2:00:00:00

L3 Top Talkers:

L4 Top Talkers:

Internal Top Talkers:
1366 Interface TwentyFiveGigE1/0/1
4096 CPU Queue Layer2 control protocols
```

# show platform software fed switch punt rates interfaces

To display the overall statistics of punt rate for all the interfaces, use the **show platform software fed switch punt rates interfaces** command in privileged EXEC mode.

**show platform software fed switch** {*switch-number* | **active** | **standby**} **punt rates interfaces**[**interface-id**]

## Syntax Description

| | |
|---|---|
| **switch**{*switch-number* \| **active** \| **standby**} | Displays information about the switch. You have the following options:<br>• *switch-number.*<br>• **active** —Displays information relating to the active switch.<br>• **standby**—Displays information relating to the standby switch, if available.<br>**Note** This keyword is not supported. |
| **punt** | Specifies the punt informtion. |
| **rates** | Specifies the rate at which the packets are punted. |
| **interfaces**[**interface-id**] | (Optional) Displays the overall statistics for an interface and also the per-queue configuration for the interface at an interval of 10 seconds. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Gibraltar 16.10.1 | This command was introduced. |

## Usage Guidelines

The output displays the punt rates in packets per second at intervals of 10 seconds, 1 minute and 5 minutes.

### Example

The following is sample output from the **show platform software fed switch active punt rates interfaces** command for all the interfaces.

```
Device#show plataform software fed switch active punt rates interfaces

Punt Rate on Interfaces Statistics

Packets per second averaged over 10 seconds, 1 min and 5 mins

===========================================================================================

                            |           |  Rx   |  Rx   |  Rx   |  Drop |  Drop  |  Drop
 Interface Name             |  IF_ID    |  10s  |  1min |  5min |  10s  |  1min  |  5min
```

```
==========================================================================================
 Vlan3                        0x00000034    1000    1000     520       0       0       0
------------------------------------------------------------------------------------------
```

The table below describes the significant fields shown in the display.

*Table 8: show platform software fed switch active punt rates interfaces Field Descriptions*

| Field | Description |
|---|---|
| Interface Name | Name of the physical interface. |
| IF_ID | ID of the physical interface. |
| Rx | The per second rate at which the packets are received in 10s, 1 minute and 5 minutes. |
| Drop | The per second rate at which the packets are dropped in 10s, 1 minute and 5 minutes. |

The following is sample output from the **show platform software fed switch active punt rates interfaces** *interface-id* command for a specific interface.

```
Device#show platform software fed switch active punt rates interfaces 0x31
Punt Rate on Single Interfaces Statistics

Interface : Port-channel1 [if_id: 0x31]

  Received                         Dropped
  --------                         -------
   Total           : 29617           Total            : 0
   10 sec average : 0                10 sec average  : 0
    1 min average : 0                 1 min average  : 0
    5 min average : 0                 5 min average  : 0

Per CPUQ punt stats on the interface (rate averaged over 10s interval)
=======================================================================
 Q  |         Queue                 |  Recv   |  Recv  |  Drop  |  Drop  |
 no |         Name                  |  Total  |  Rate  |  Total |  Rate  |
=======================================================================
 0   CPU_Q_DOT1X_AUTH                     0        0        0        0
 1   CPU_Q_L2_CONTROL                 29519        0        0        0
 2   CPU_Q_FORUS_TRAFFIC                  0        0        0        0
 3   CPU_Q_ICMP_GEN                       0        0        0        0
 4   CPU_Q_ROUTING_CONTROL                0        0        0        0
 5   CPU_Q_FORUS_ADDR_RESOLUTION          0        0        0        0
 6   CPU_Q_ICMP_REDIRECT                  0        0        0        0
 7   CPU_Q_INTER_FED_TRAFFIC              0        0        0        0
 8   CPU_Q_L2LVX_CONTROL_PKT              0        0        0        0
 9   CPU_Q_EWLC_CONTROL                   0        0        0        0
 10  CPU_Q_EWLC_DATA                      0        0        0        0
 11  CPU_Q_L2LVX_DATA_PKT                 0        0        0        0
 12  CPU_Q_BROADCAST                      0        0        0        0
 13  CPU_Q_LEARNING_CACHE_OVFL            0        0        0        0
 14  CPU_Q_SW_FORWARDING                  0        0        0        0
 15  CPU_Q_TOPOLOGY_CONTROL              98        0        0        0
 16  CPU_Q_PROTO_SNOOPING                 0        0        0        0
 17  CPU_Q_DHCP_SNOOPING                  0        0        0        0
 18  CPU_Q_TRANSIT_TRAFFIC                0        0        0        0
 19  CPU_Q_RPF_FAILED                     0        0        0        0
 20  CPU_Q_MCAST_END_STATION_SERVICE        0         0         0         0
 21  CPU_Q_LOGGING                          0         0         0         0
 22  CPU_Q_PUNT_WEBAUTH                     0         0         0         0
 23  CPU_Q_HIGH_RATE_APP                    0         0         0         0
 24  CPU_Q_EXCEPTION                        0         0         0         0
 25  CPU_Q_SYSTEM_CRITICAL                  0         0         0         0
 26  CPU_Q_NFL_SAMPLED_DATA                 0         0         0         0
 27  CPU_Q_LOW_LATENCY                      0         0         0         0
 28  CPU_Q_EGR_EXCEPTION                    0         0         0         0
 29  CPU_Q_FSS                              0         0         0         0
 30  CPU_Q_MCAST_DATA                       0         0         0         0
 31  CPU_Q_GOLD_PKT                         0         0         0         0

--------------------------------------------------------------------------
```

The table below describes the significant fields shown in the display.

*Table 9:* ***show platform software fed switch punt rates interfaces*** *interface-id* *Field Descriptions*

| Field | Description |
|---|---|
| Queue Name | Name of the queue. |
| Recv Total | Total number of packets received. |
| Recv Rate | Per second rate at which the packets are received. |
| Drop Total | Total number of packets dropped. |
| Drop Rate | Per second rate at which the packets are dropped. |

# show platform software ilpower

To display the inline power details of all the PoE ports on the device, use the **show platform software ilpower** command in privileged EXEC mode.

**show platform software ilpower** {**details** | **port** {**GigabitEthernet** *interface-number* } | **system** *slot-number* }

## Syntax Description

| | |
|---|---|
| **details** | Displays inline power details for all the interfaces. |
| **port** | Displays inline power port configuration. |
| **GigabitEthernet** *interface-number* | The GigabitEthernet interface number. Values range from 0 to 9. |
| **system** *slot-number* | Displays inline power system configuration. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | The command was introduced. |

## Examples

The following is sample output from the **show platform software ilpower details** command:

```
Device# show platform software ilpower details
ILP Port Configuration for interface Gi1/0/1
    Initialization Done:   Yes
    ILP Supported:         Yes
    ILP Enabled:           Yes
    POST:                  Yes
    Detect On:             No
    Powered Device Detected            No
    Powered Device Class Done          No
    Cisco Powered Device:              No
    Power is On:           No
    Power Denied:          No
    Powered Device Type:               Null
    Powerd Device Class:              Null
    Power State:           NULL
    Current State:         NGWC_ILP_DETECTING_S
    Previous State:        NGWC_ILP_SHUT_OFF_S
    Requested Power in milli watts:       0
    Short Circuit Detected:                 0
    Short Circuit Count:             0
    Cisco Powerd Device Detect Count: 0
    Spare Pair mode:       0
        IEEE Detect:       Stopped
        IEEE Short:        Stopped
        Link Down:         Stopped
        Voltage sense:            Stopped
    Spare Pair Architecture:       1
    Signal Pair Power allocation in milli watts: 0
    Spare Pair Power On:   0
    Powered Device power state:        0
    Timer:
```

```
Power Good:        Stopped
Power Denied:      Stopped
Cisco Powered Device Detect:   Stopped
```

# show platform software memory

To display memory information for a specified switch, use the **show platform software memory** command in privileged EXEC mode.

**show platform software memory** [{**chunk** | **database** | **messaging**}] *process slot*

**Syntax Description**

**Syntax Description**

| | |
|---|---|
| **chunk** | (Optional) Displays chunk memory information for the specified process. |
| **database** | (Optional) Displays database memory information for the specified process. |
| **messaging** | (Optional) Displays messaging memory information for the specified process.<br>The information displayed is for internal debugging purposes only. |

| *process* |
|---|

Level that is being set. Options include:

- **bt-logger**—The Binary-Tracing Logger process.
- **btrace-manager**—The Btrace Manager process.
- **chassis-manager**—The Chassis Manager process.
- **cli-agent**—The CLI Agent process.
- **cmm**—The CMM process.
- **dbm**—The Database Manager process.
- **dmiauthd**—The DMI Authentication Daemon process.
- **emd**—The Environmental Monitoring process.
- **fed**—The Forwarding Engine Driver process.
- **forwarding-manager**—The Forwarding Manager process.
- **geo**—The Geo Manager process.
- **gnmi**—The GNMI process.
- **host-manager**—The Host Manager process.
- **interface-manager**—The Interface Manager process.
- **iomd**—The Input/Output Module daemon (IOMd) process.
- **ios**—The IOS process.
- **iox-manager**—The IOx Manager process.
- **license-manager**—The License Manager process.
- **logger**—The Logging Manager process.
- **mdt-pubd**—The Model Defined Telemetry Publisher process.
- **ndbman**—The Netconf DataBase Manager process.
- **nesd**—The Network Element Synchronizer Daemon process.
- **nginx**—The Nginx Webserver process.
- **nif_mgr**—The NIF Manager process.
- **platform-mgr**—The Platform Manager process.
- **pluggable-services**—The Pluggable Services process.
- **replication-mgr**—The Replication Manager process.
- **shell-manager**—The Shell Manager process.
- **sif**—The Stack Interface (SIF) Manager process.
- **smd**—The Session Manager process.
- **stack-mgr**—The Stack Manager process.

| | |
|---|---|
| | • **syncfd**—The SyncmDaemon process.<br>• **table-manager**—The Table Manager Server.<br>• **thread-test**—The Multithread Manager process.<br>• **virt-manager**—The Virtualization Manager process. |
| *slot* | Hardware slot where the process for which the level is set, is running. Options include:<br>• *number*—Number of the SIP slot of the hardware module where the level is set. For instance, if you want to specify the SIP in SIP slot 2 of the switch, enter 2.<br>• *SIP-slot / SPA-bay*—Number of the SIP switch slot and the number of the shared port adapter (SPA) bay of that SIP. For instance, if you want to specify the SPA in bay 2 of the SIP in switch slot 3, enter 3/2.<br>• **F0**—The Embedded Service Processor slot 0.<br>• **FP active**—The active Embedded Service Processor.<br>• **R0**—The route processor in slot 0.<br>• **RP active**—The active route processor.<br>• **RP standby**—The standby route processor.<br>• **switch** <*number*> —The switch, with its number specified.<br>• **switch active**—The active switch.<br>• **switch standby**—The standby switch.<br>  • *number*—Number of the SIP slot of the hardware module where the level is set. For instance, if you want to specify the SIP in SIP slot 2 of the switch, enter 2.<br>  • *SIP-slot / SPA-bay*—Number of the SIP switch slot and the number of the shared port adapter (SPA) bay of that SIP. For instance, if you want to specify the SPA in bay 2 of the SIP in switch slot 3, enter 3/2.<br>  • **F0**—The Embedded Service Processor in slot 0.<br>  • **FP active**—The active Embedded Service Processor.<br>  • **R0**—The route processor in slot 0.<br>  • **RP active**—The active route processor. |

### Command Default

No default behavior or values.

### Command Modes

Privileged EXEC (#)

## Command History

Command History

| Release | Modificatio |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This comm |

The following is a sample output displaying the abbreviated (brief keyword) memory information for the Forwarding Manager process for Cisco Catalyst 9000 Series ESP slot 0:

```
Device# show platform software memory forwarding-manager switch 1 fp active brief

  module                  allocated     requested     allocs        frees
  ------------------------------------------------------------------------------
  Summary                 5702540       5619788       121888        116716
  AOM object              1920374       1920310       4             0
  AOM links array         880379        880315        4             0
  smc_message             819575        819511        4             0
  AOM update state        640380        640316        4             0
  dpidb-config            208776        203544        351           24
  fman-infra-avl          178016        153680        1521          0
  AOM batch               152373        152309        4             0
  AOM asynchronous conte  128388        128324        4             0
  AOM basic data          124824        124760        5             1
  eventutil               118939        118299        50            10
  AOM tree node           96465         96385         5             0
  AOM tree root           72377         72313         4             0
  acl                     36090         31914         504           243
  fman-infra-ipc          35326         24366         115097        114412
  AOM uplink update node  32386         32322         4             0
  unknown                 30528         23808         424           4
  uipeer                  27232         27152         5             0
  fman-infra-qos          26872         24712         164           29
  cce-class               19427         15411         251           0
  l2 control protocol     15472         12896         325           164
  fman-infra-cce          15272         13576         106           0
  smc_channel             15223         15159         4             0
  unknown                 14208         8736          447           105
  chunk                   12513         12033         33            3
  cce-bind                8496          7552          82            23
  MATM mac entry          8040          5928          544           412
  adj                     7064          6312          157           110
  route-pfx               6116          5412          157           113
  Filter_rules            4912          4896          1             0
  fman-infra-dpidb        4130          2338          112           0
  SMC Buffer              3794          3202          43            6
  urpf-list               3028          2100          85            27
  lookup                  2480          2160          30            10
  MATM mac table          2432          1600          148           96
  cdllib                  1688          1672          1             0
  route-tbl               1600          1264          21            0
  FNF Flowdef             1492          1460          3             1
  acl-ref                 1120          1024          8             2
  cgm-lib                 1120          880           410           395
  pbr_if_cfg              1088          976           205           198
  FNF Monitor             1048          1032          1             0
  pbr_routemap            960           864           18            12
!
!
!
```

The following table describes the significant fields shown in the display.

*Table 10: show platform software memory brief Field Descriptions*

| Field | Description |
|---|---|
| module | Name of submodule. |
| allocated | Memory, allocated in bytes. |
| requested | Number of bytes requested by application. |
| allocs | Number of discrete allocation event attempts. |
| frees | Number of free events. |

# show platform software process list

To display the list of running processes on a platform, use the **show platform software process list** command in privileged EXEC mode.

**show platform software process list switch** {*switch-number* | **active** | **standby**} {**0** | **F0** | **R0**} [{**name** *process-name* | **process-id** *process-ID* | **sort memory** | **summary**}]

## Syntax Description

| | |
|---|---|
| **switch** *switch-number* | Displays information about the switch. Valid values for *switch-number* argument are from 0 to 9. |
| **active** | Displays information about the active instance of the switch. |
| **standby** | Displays information about the standby instance of the switch. |
| **0** | Displays information about the shared port adapters (SPA) Interface Processor slot 0. |
| **F0** | Displays information about the Embedded Service Processor (ESP) slot 0. |
| **R0** | Displays information about the Route Processor (RP) slot 0. |
| **name** *process-name* | (Optional) Displays information about the specified process. Enter the process name. |
| **process-id** *process-ID* | (Optional) Displays information about the specified process ID. Enter the process ID. |
| **sort** | (Optional) Displays information sorted according to processes. |
| **memory** | (Optional) Displays information sorted according to memory. |
| **summary** | (Optional) Displays a summary of the process memory of the host device. |

## Command Modes

Privileged EXE (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | The command was introduced. |

## Examples

The following is sample output from the **show platform software process list switch active R0** command:

```
Switch# show platform software process list switch active R0 summary

Total number of processes: 278
  Running          : 2
  Sleeping         : 276
  Disk sleeping    : 0
  Zombies          : 0
  Stopped          : 0
  Paging           : 0

  Up time          : 8318
  Idle time        : 0
  User time        : 216809
  Kernel time      : 78931

  Virtual memory   : 12933324800
  Pages resident   : 634061
  Major page faults: 2228
  Minor page faults: 3491744

  Architecture     : mips64
  Memory (kB)
    Physical       : 3976852
    Total          : 3976852
    Used           : 2766952
    Free           : 1209900
    Active         : 2141344
    Inactive       : 1589672
    Inact-dirty    : 0
    Inact-clean    : 0
    Dirty          : 4
    AnonPages      : 1306800
    Bounce         : 0
    Cached         : 1984688
    Commit Limit   : 1988424
    Committed As   : 3358528
    High Total     : 0
    High Free      : 0
    Low Total      : 3976852
    Low Free       : 1209900
    Mapped         : 520528
    NFS Unstable   : 0
    Page Tables    : 17328
    Slab           : 0
    VMmalloc Chunk : 1069542588
    VMmalloc Total : 1069547512
    VMmalloc Used  : 2588
    Writeback      : 0
    HugePages Total: 0
    HugePages Free : 0
    HugePages Rsvd : 0
    HugePage Size  : 2048

  Swap (kB)
    Total          : 0
    Used           : 0
    Free           : 0
    Cached         : 0

  Buffers (kB)     : 439528

  Load Average
    1-Min          : 1.13
    5-Min          : 1.18
    15-Min         : 0.92
```

The following is sample output from the **show platform software process list switch active R0** command:

```
# show platform software process list switch active R0
Name                     Pid    PPid  Group Id  Status    Priority  Size
------------------------------------------------------------------------------
```

```
systemd                    1       0         1  S            20  4876
kthreadd                   2       0         0  S            20  0
ksoftirqd/0                3       2         0  S            20  0
kworker/0:0H               5       2         0  S             0  0
rcu_sched                  7       2         0  S            20  0
rcu_bh                     8       2         0  S            20  0
migration/0                9       2         0  S    4294967196  0
watchdog/0                10       2         0  S    4294967196  0
watchdog/1                11       2         0  S    4294967196  0
migration/1               12       2         0  S    4294967196  0
ksoftirqd/1               13       2         0  S            20  0
kworker/1:0H              15       2         0  S             0  0
watchdog/2                16       2         0  S    4294967196  0
migration/2               17       2         0  S    4294967196  0
ksoftirqd/2               18       2         0  S            20  0
kworker/2:0H              20       2         0  S             0  0
watchdog/3                21       2         0  S    4294967196  0
migration/3               22       2         0  S    4294967196  0
ksoftirqd/3               23       2         0  S            20  0
kworker/3:0               24       2         0  S            20  0
kworker/3:0H              25       2         0  S             0  0
kdevtmpfs                 26       2         0  S            20  0
netns                     27       2         0  S             0  0
perf                      28       2         0  S             0  0
khungtaskd                29       2         0  S            20  0
writeback                 30       2         0  S             0  0
ksmd                      31       2         0  S            25  0
khugepaged                32       2         0  S            39  0
crypto                    33       2         0  S             0  0
bioset                    34       2         0  S             0  0
kblockd                   35       2         0  S             0  0
ata_sff                   36       2         0  S             0  0
rpciod                    37       2         0  S             0  0
kswapd0                   63       2         0  S            20  0
vmstat                    64       2         0  S             0  0
fsnotify_mark             65       2         0  S            20  0
nfsiod                    66       2         0  S             0  0
.
.
.
```

The table below describes the significant fields shown in the displays.

***Table 11: show platform software process list Field Descriptions***

| Field | Description |
|---|---|
| Name | Displays the command name associated with the process. Different threads in the same process may have different command values. |
| Pid | Displays the process ID that is used by the operating system to identify and keep track of the processes. |
| PPid | Displays process ID of the parent process. |
| Group Id | Displays the group ID |
| Status | Displays the process status in human readable form. |

| Field | Description |
|---|---|
| Priority | Displays the negated scheduling priority. |
| Size | Prior to Cisco IOS XE Gibraltar 16.10.1:<br>Displays Virtual Memory size.<br>From Cisco IOS XE Gibraltar 16.10.1 onwards:<br>Displays the Resident Set Size (RSS) that shows how much memory is allocated to that process in the RAM. |

# show platform software process memory

To display the amount of memory used by each system process, use the **show platform software process memory** command in privileged EXEC mode.

**show platform process memory**
**switch** { *switch-number* | **active** | **standby** } { **0** | **F0** | **FP** | **R0** } { **all** [ **sorted** | **virtual** [ **sorted** ] ] | **name** *process-name* { **maps** | **smaps** [ **summary** ] } | **process-id** *process-id* { **maps** | **smaps** [ **summary** ] } }

## Syntax Description

| | |
|---|---|
| **switch** *switch-number* | Displays information about the switch. Enter the switch number. |
| **active** | Specifies the active instance of the device. |
| **standby** | Specifies the standby instance of the device. |
| **0** | Specifies the Shared Port Adapter (SPA) Interface Processor slot 0. |
| **F0** | Specifies the Embedded Service Processor (ESP) slot 0. |
| **FP** | Specifies the Embedded Service Processor (ESP). |
| **R0** | Specifies the Route Processor (RP) slot 0. |
| **all** | Lists all processes. |
| **sorted** | (Optional) Sorts the output based on Resident Set Size (RSS). |
| **virtual** | (Optional) Specifies virtual memory. |
| **name** *process-name* | Specifies a process name. |
| **maps** | Specifies the memory maps of a process. |
| **smaps summary** | Specifies the smaps summary of a process. |
| **process-id** *process-id* | Specifies a process identifier. |

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Gibraltar 16.10.1 | This command was introduced. |

## Command Modes

Privileged EXEC(#)

### Examples:

The following is a sample output from the **show platform software process memory active R0 all** command:

```
Device# show platform software process memory switch active R0 all

   Pid       RSS        PSS       Heap    Shared   Private                Name
--------------------------------------------------------------------------------
     1      4876       3229       1064      1808      3068             systemd
   118      3184       1327        132      2352       832     systemd-journal
   159      3008       1191        396      1996      1012       systemd-udevd
   407      3192       1262        132      2196       996         dbus-daemon
  3406      4772       3064        264      1940      2832            virtlogd
  3411      5712       3474       2964      2344      3368         droputil.sh
  3416      2588        358        132      2336       252         libvirtd.sh
  3420      5708       3484       2976      2308      3400        reflector.sh
  3424      1804        263        132      1632       172              xinetd
  3425       964        118        132       872        92               sleep
  3434      3060        844        528      2304       756              oom.sh
  3442      2068        606        132      1604       464             rpcbind
  3485      2380        845        132      1636       744           rpc.statd
  3486      1632        338        132      1348       284     boothelper_evt.
  3493      1136        156        132      1004       132         inotifywait
  3504      2048        753        132      1372       676          rpc.mountd
  3584      2868        620         36      2384       484               rotee
  3649      1032        116        132       944        88               sleep
  3705      2784        613         36      2296       488               rotee
  3718      2856        610         36      2376       480               rotee
  3759      1292        184        132      1136       156         inotifywait
  3787      4256       2040       1640      2300      1956           iptbl.sh
  3894      2948        637         36      2460       488               rotee
  4017      1380        175        132      1236       144         inotifywait
  4866      1820        287        132      1624       196              xinetd
  5887      1692        257        132      1508       184              xinetd
  5891      7248       4984       4584      2348      4900     rollback_timer.
  5893      1764        257        132      1588       176              xinetd
  6031      2804        601         36      2332       472               rotee
  6037      1228        163        132      1092       136         inotifywait
  6077      4736       3389       2992      1368      3368             psvp.sh
  6115      1620        476         36      1152       468               rotee
  6122       624        149        132       480       144         inotifywait
  6127      5440       4077       3680      1384      4056              pvp.sh
  6165      1736        592         36      1152       584               rotee
  6245       624        149        132       480       144         inotifywait
  6353      2592       1260        924      1352      1240             pman.sh
  6470      1632        488         36      1152       480               rotee
  6499      2588       1262        924      1348      1240             pman.sh
  6666      1640        496         36      1152       488               rotee
  6718      2584       1258        800      1348      1236             pman.sh
  6736      8360       7020       6640      1360      7000     auto_upgrade_cl
  6909      1636        492         36      1152       484               rotee
  6955      2588       1262        928      1348      1240             pman.sh
  7029      2196        679         40      1552       644     auto_upgrade_se
  7149      1636        492         36      1152       484               rotee
  7224     13200       4595         48      9368      3832           bt_logger
  7295      2588       1262        800      1348      1240             pman.sh
.
.
.
```

The table below describes the significant fields shown in the displays.

*Table 12: show platform software process memory Field Descriptions*

| Field | Description |
|---|---|
| PID | Displays the process ID that is used by the operating system to identify and keep track of the processes. |
| RSS | Displays the Resident Set Size (in kilobytes (KB)) that shows how much memory is allocated to that process in the RAM. |
| PSS | Displays the Proportional Set Size of a process. This is the count of pages it has in memory, where each page is divided by the number of processes sharing it. |
| Heap | Displays where all user-allocated memory is located. |
| Shared | Shared clean + Shared dirty |
| Private | Private clean + Private dirty |
| Name | Displays the command name associated with the process. Different threads in the same process may have different command values. |

# show platform software process slot switch

To display platform software process switch information, use the **show platform software process slot switch** command in privileged EXEC mode.

**show platform software process slot switch** {*switch-number* | **active** | **standby**} {**0** | **F0** | **R0**} **monitor** [{**cycles** *no-of-times* [{**interval** *delay* [{**lines** *number*}]}]}]

## Syntax Description

| | |
|---|---|
| *switch-number* | Switch number. |
| **active** | Specifies the active instance. |
| **standby** | Specifies the standby instance. |
| **0** | Specifies the shared port adapter (SPA) interface processor slot 0. |
| **F0** | Specifies the Embedded Service Processor (ESP) slot 0. |
| **R0** | Specifies the Route Processor (RP) slot 0. |
| **monitor** | Monitors the running processes. |
| **cycles** *no-of-tmes* | (Optional) Sets the number of times to run monitor command. Valid values are from 1 to 4294967295. The default is 5. |
| **interval** *delay* | (Optional) Sets a delay after each . Valid values are from 0 to 300. The default is 3. |
| **lines** *number* | (Optional) Sets the number of lines of output displayed. Valid values are from 0 to 512. The default is 0. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

The output of the **show platform software process slot switch** and **show processes cpu platform monitor location** commands display the output of the Linux **top** command. The output of these commands display Free memory and Used memory as displayed by the Linux **top** command. The values displayed for the Free memory and Used memory by these commands do not match the values displayed by the output of other platform-memory related CLIs.

## Examples

The following is sample output from the **show platform software process slot monitor** command:

```
Switch# show platform software process slot switch active R0 monitor
```

```
top - 00:01:52 up 1 day, 11:20,  0 users,  load average: 0.50, 0.68, 0.83
Tasks: 311 total,   2 running, 309 sleeping,   0 stopped,   0 zombie
Cpu(s):  7.4%us,  3.3%sy,  0.0%ni, 89.2%id,  0.0%wa,  0.0%hi,  0.1%si,  0.0%st
Mem:   3976844k total,  3955036k used,    21808k free,   419312k buffers
Swap:        0k total,        0k used,        0k free,  1946764k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 5693 root      20   0  3448 1368  912 R    7  0.0   0:00.07 top
17546 root      20   0 2044m 244m  79m S    7  6.3 186:49.08 fed main event
18662 root      20   0 1806m 678m 263m S    5 17.5 215:32.38 linux_iosd-imag
30276 root      20   0  171m  42m  33m S    5  1.1 125:06.77 repm
17835 root      20   0  935m  74m  63m S    4  1.9  82:28.31 sif_mgr
18534 root      20   0  182m 150m  10m S    2  3.9   8:12.08 smand
    1 root      20   0  8440 4740 2184 S    0  0.1   0:09.52 systemd
    2 root      20   0     0    0    0 S    0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S    0  0.0   0:02.86 ksoftirqd/0
    5 root       0 -20     0    0    0 S    0  0.0   0:00.00 kworker/0:0H
    7 root      RT   0     0    0    0 S    0  0.0   0:01.44 migration/0
    8 root      20   0     0    0    0 S    0  0.0   0:00.00 rcu_bh
    9 root      20   0     0    0    0 S    0  0.0   0:23.08 rcu_sched
   10 root      20   0     0    0    0 S    0  0.0   0:58.04 rcuc/0
   11 root      20   0     0    0    0 S    0  0.0  21:35.60 rcuc/1
   12 root      RT   0     0    0    0 S    0  0.0   0:01.33 migration/1
```

## Related Commands

| Command | Description |
|---|---|
| **show processes cpu platform monitor location** | Displays information about the CPU utilization of the IOS-XE processes. |

# show platform software status control-processor

To display platform software control-processor status, use the **show platform software status control-processor** command in privileged EXEC mode.

**show platform software status control-processor** [{**brief**}]

## Syntax Description

| | |
|---|---|
| **brief** | (Optional) Displays a summary of the platform control-processor status. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Examples

The following is sample output from the **show platform memory software status control-processor** command:

```
Switch# show platform software status control-processor

2-RP0: online, statistics updated 7 seconds ago
Load Average: healthy
  1-Min: 1.00, status: healthy, under 5.00
  5-Min: 1.21, status: healthy, under 5.00
  15-Min: 0.90, status: healthy, under 5.00
Memory (kb): healthy
  Total: 3976852
  Used: 2766284 (70%), status: healthy
  Free: 1210568 (30%)
  Committed: 3358008 (84%), under 95%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
  User:  4.40, System:  1.70, Nice:  0.00, Idle: 93.80
  IRQ:  0.00, SIRQ:  0.10, IOwait:  0.00
CPU1: CPU Utilization (percentage of time spent)
  User:  3.80, System:  1.20, Nice:  0.00, Idle: 94.90
  IRQ:  0.00, SIRQ:  0.10, IOwait:  0.00
CPU2: CPU Utilization (percentage of time spent)
  User:  7.00, System:  1.10, Nice:  0.00, Idle: 91.89
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00
CPU3: CPU Utilization (percentage of time spent)
  User:  4.49, System:  0.69, Nice:  0.00, Idle: 94.80
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00

3-RP0: unknown, statistics updated 2 seconds ago
Load Average: healthy
  1-Min: 0.24, status: healthy, under 5.00
  5-Min: 0.27, status: healthy, under 5.00
  15-Min: 0.32, status: healthy, under 5.00
Memory (kb): healthy
  Total: 3976852
  Used: 2706768 (68%), status: healthy
  Free: 1270084 (32%)
  Committed: 3299332 (83%), under 95%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
```

```
  User:  4.50, System:  1.20, Nice:  0.00, Idle: 94.20
  IRQ:  0.00, SIRQ:  0.10, IOwait:  0.00
CPU1: CPU Utilization (percentage of time spent)
  User:  5.20, System:  0.50, Nice:  0.00, Idle: 94.29
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00
CPU2: CPU Utilization (percentage of time spent)
  User:  3.60, System:  0.70, Nice:  0.00, Idle: 95.69
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00
CPU3: CPU Utilization (percentage of time spent)
  User:  3.00, System:  0.60, Nice:  0.00, Idle: 96.39
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00

4-RP0: unknown, statistics updated 2 seconds ago
Load Average: healthy
  1-Min: 0.21, status: healthy, under 5.00
  5-Min: 0.24, status: healthy, under 5.00
  15-Min: 0.24, status: healthy, under 5.00
Memory (kb): healthy
  Total: 3976852
  Used: 1452404 (37%), status: healthy
  Free: 2524448 (63%)
  Committed: 1675120 (42%), under 95%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
  User:  2.30, System:  0.40, Nice:  0.00, Idle: 97.30
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00
CPU1: CPU Utilization (percentage of time spent)
  User:  4.19, System:  0.69, Nice:  0.00, Idle: 95.10
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00
CPU2: CPU Utilization (percentage of time spent)
  User:  4.79, System:  0.79, Nice:  0.00, Idle: 94.40
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00
CPU3: CPU Utilization (percentage of time spent)
  User:  2.10, System:  0.40, Nice:  0.00, Idle: 97.50
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00

9-RP0: unknown, statistics updated 4 seconds ago
Load Average: healthy
  1-Min: 0.20, status: healthy, under 5.00
  5-Min: 0.35, status: healthy, under 5.00
  15-Min: 0.35, status: healthy, under 5.00
Memory (kb): healthy
  Total: 3976852
  Used: 1451328 (36%), status: healthy
  Free: 2525524 (64%)
  Committed: 1675932 (42%), under 95%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
  User:  1.90, System:  0.50, Nice:  0.00, Idle: 97.60
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00
CPU1: CPU Utilization (percentage of time spent)
  User:  4.39, System:  0.19, Nice:  0.00, Idle: 95.40
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00
CPU2: CPU Utilization (percentage of time spent)
  User:  5.70, System:  1.00, Nice:  0.00, Idle: 93.30
  IRQ:  0.00, SIRQ:  0.00, IOwait:  0.00
CPU3: CPU Utilization (percentage of time spent)
  User:  1.30, System:  0.60, Nice:  0.00, Idle: 98.00
  IRQ:  0.00, SIRQ:  0.10, IOwait:  0.00
```

The following is sample output from the **show platform memory software status control-processor brief** command:

```
Switch# show platform software status control-processor brief

Load Average
 Slot  Status  1-Min  5-Min 15-Min
2-RP0 Healthy   1.10   1.21   0.91
3-RP0 Healthy   0.23   0.27   0.31
4-RP0 Healthy   0.11   0.21   0.22
9-RP0 Healthy   0.10   0.30   0.34

Memory (kB)
 Slot  Status    Total     Used (Pct)     Free (Pct) Committed (Pct)
2-RP0 Healthy  3976852  2766956 (70%)  1209896 (30%)   3358352 (84%)
3-RP0 Healthy  3976852  2706824 (68%)  1270028 (32%)   3299276 (83%)
4-RP0 Healthy  3976852  1451888 (37%)  2524964 (63%)   1675076 (42%)
9-RP0 Healthy  3976852  1451580 (37%)  2525272 (63%)   1675952 (42%)

CPU Utilization
 Slot  CPU   User System   Nice   Idle    IRQ   SIRQ IOwait
2-RP0    0   4.10   2.00   0.00  93.80   0.00   0.10   0.00
         1   4.60   1.00   0.00  94.30   0.00   0.10   0.00
         2   6.50   1.10   0.00  92.40   0.00   0.00   0.00
         3   5.59   1.19   0.00  93.20   0.00   0.00   0.00
3-RP0    0   2.80   1.20   0.00  95.90   0.00   0.10   0.00
         1   4.49   1.29   0.00  94.20   0.00   0.00   0.00
         2   5.30   1.60   0.00  93.10   0.00   0.00   0.00
         3   5.80   1.20   0.00  93.00   0.00   0.00   0.00
4-RP0    0   1.30   0.80   0.00  97.89   0.00   0.00   0.00
         1   1.30   0.20   0.00  98.50   0.00   0.00   0.00
         2   5.60   0.80   0.00  93.59   0.00   0.00   0.00
         3   5.09   0.19   0.00  94.70   0.00   0.00   0.00
9-RP0    0   3.99   0.69   0.00  95.30   0.00   0.00   0.00
         1   2.60   0.70   0.00  96.70   0.00   0.00   0.00
         2   4.49   0.89   0.00  94.60   0.00   0.00   0.00
         3   2.60   0.20   0.00  97.20   0.00   0.00   0.00
```

# show platform software thread list

To display the list of threads on a platform, use the **show platform software thread list** command in privileged EXEC mode.

**show platform software thread list switch** { *switch-number* | **active** | **standby** } { **0** | **F0** | **FP active** | **R0** } **pname** { **cdman** | **vidman** | **all** } **tname** { **main** | **pktio** | **rt** | **all** }

## Syntax Description

| | |
|---|---|
| **switch** *switch-number* | Displays information about the switch. Enter the switch number. |
| **active** | Specifies the active instance of the device. |
| **standby** | Specifies standby instance of the device. |
| **0** | Specifies the Shared Port Adapter (SPA) Interface Processor slot 0. |
| **F0** | Specifies the Embedded Service Processor (ESP) slot 0. |
| **FP active** | Specifies the active instance of Embedded Service Processor (ESP). |
| **R0** | Specifies the Route Processor (RP) slot 0. |
| **pname** | Specifies a process name. The possible values are **cdman**, **vidman**, and **all**. |
| **tname** | Specifies a thread name. The possible values are **main**, **pktio**, **rt**, and **all**. |

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Gibraltar 16.10.1 | This command was introduced. |

## Command Modes

Privileged EXEC(#)

### Examples:

The following is sample output from the **show platform software thread list switch active R0 pname cdman tname all** command:

```
Device# show platform software thread list switch active R0 pname cdman tname all

Name           Tid    PPid  Group Id  Core    Vcswch   Nvcswch  Status    Priority
TIME+  Size
------------------------------------------------------------------------------------------
cdman         8407    7295      8407     1         0         0  S              20
12309  36976
```

The table below describes the significant fields shown in the displays.

*Table 13: show platform software thread list Field Descriptions*

| Field | Description |
|---|---|
| Name | Displays the command name associated with the process. Different threads in the same process may have different command values. |
| Tid | Displays the process ID. |
| PPid | Displays the process ID of the parent process. |
| Group Id | Displays the group ID. |
| Core | Displays processor information. |
| Vcswch | Displays the number of voluntary context switches. |
| Nvcswch | Displays the number of non-voluntary context switches. |
| Status | Displays the process status in human readable form. |
| Priority | Displays the negated scheduling priority. |
| TIME+ | Displays the time since the start of the process. |
| Size | Displays the Resident Set Size (in kilobytes (KB)) that shows how much memory is allocated to that process in the RAM. |

# show platform usb status

To display the status of the USB ports on a device, use the **show platform usb status** command in Privileged EXEC mode.

**show platform usb status**

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Bengaluru 17.5.1 | This command was introduced. |

## Examples

The following is a sample output of the **show platform usb status** command:

```
Device> enable
Device# show platform usb status
USB Disabled
```

# show processes cpu platform

To display information about the CPU utilization of the IOS-XE processes, use the **show processes cpu platform** command in privileged EXEC mode.

**show processes cpu platform** [ [ **sorted** [ **1min** | **5min** | **5sec** ] ] **location switch** { *switch-number* | **active** | **standby** } { **F0** | **FP active** | **R0** | **RP active** } ]

## Syntax Description

| | |
|---|---|
| **sorted** | (Optional) Displays output sorted based on percentage of CPU usage on a platform. |
| **1min** | (Optional) Sorts based on 1 minute intervals. |
| **5min** | (Optional) Sorts based on 5 minute intervals. |
| **5sec** | (Optional) Sorts based on 5 second intervals. |
| **location** | Specifies the Field Replaceable Unit (FRU) location. |
| **switch** *switch-number* | Displays information about the switch. Enter the switch number. |
| **active** | Specifies the active instance of the device. |
| **standby** | Specifies the standby instance of the device. |
| **F0** | Specifies the Embedded Service Processor (ESP) slot 0. |
| **FP active** | Specifies active instances on the Embedded Service Processor (ESP). |
| **R0** | Specifies the Route Processor (RP) slot 0. |
| **RP active** | Specifies active instances on the Route Processor (RP). |

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Gibraltar 16.10.1 | This command was introduced. |

## Command Modes

Privileged EXEC (#)

### Examples:

The following is sample output from the **show processes cpu platform** command:

```
Device# show processes cpu platform

CPU utilization for five seconds:  1%, one minute:  3%, five minutes:  2%
Core 0: CPU utilization for five seconds:  2%, one minute:  2%, five minutes:  2%
Core 1: CPU utilization for five seconds:  2%, one minute:  1%, five minutes:  1%
Core 2: CPU utilization for five seconds:  3%, one minute:  1%, five minutes:  1%
Core 3: CPU utilization for five seconds:  2%, one minute:  5%, five minutes:  2%
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
--------------------------------------------------------------------------------
     1       0      0%      0%      0%  S             4876  systemd
```

```
      2        0      0%      0%      0%   S                0  kthreadd
      3        2      0%      0%      0%   S                0  ksoftirqd/0
      5        2      0%      0%      0%   S                0  kworker/0:0H
      7        2      0%      0%      0%   S                0  rcu_sched
      8        2      0%      0%      0%   S                0  rcu_bh
      9        2      0%      0%      0%   S                0  migration/0
     10        2      0%      0%      0%   S                0  watchdog/0
     11        2      0%      0%      0%   S                0  watchdog/1
     12        2      0%      0%      0%   S                0  migration/1
     13        2      0%      0%      0%   S                0  ksoftirqd/1
     15        2      0%      0%      0%   S                0  kworker/1:0H
     16        2      0%      0%      0%   S                0  watchdog/2
     17        2      0%      0%      0%   S                0  migration/2
     18        2      0%      0%      0%   S                0  ksoftirqd/2
     20        2      0%      0%      0%   S                0  kworker/2:0H
     21        2      0%      0%      0%   S                0  watchdog/3
     22        2      0%      0%      0%   S                0  migration/3
     23        2      0%      0%      0%   S                0  ksoftirqd/3
     24        2      0%      0%      0%   S                0  kworker/3:0
     25        2      0%      0%      0%   S                0  kworker/3:0H
     26        2      0%      0%      0%   S                0  kdevtmpfs
     27        2      0%      0%      0%   S                0  netns
     28        2      0%      0%      0%   S                0  perf
     29        2      0%      0%      0%   S                0  khungtaskd
     30        2      0%      0%      0%   S                0  writeback
     31        2      7%      8%      8%   S                0  ksmd
     32        2      0%      0%      0%   S                0  khugepaged
     33        2      0%      0%      0%   S                0  crypto
     34        2      0%      0%      0%   S                0  bioset
     35        2      0%      0%      0%   S                0  kblockd
     36        2      0%      0%      0%   S                0  ata_sff
     37        2      0%      0%      0%   S                0  rpciod
     63        2      0%      0%      0%   S                0  kswapd0
     64        2      0%      0%      0%   S                0  vmstat
     65        2      0%      0%      0%   S                0  fsnotify_mark
.
.
.
```

The following is sample output from the **show processes cpu platform sorted 5min location switch 5 R0**

```
Device# show processes cpu platform sorted 5min location switch 5 R0

CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 0: CPU utilization for five seconds:  1%, one minute:  1%, five minutes:  1%
Core 1: CPU utilization for five seconds:  1%, one minute:  1%, five minutes:  1%
Core 2: CPU utilization for five seconds:  1%, one minute:  1%, five minutes:  1%
Core 3: CPU utilization for five seconds:  2%, one minute:  2%, five minutes:  1%
Core 4: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 5: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 6: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 7: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
--------------------------------------------------------------------------------
 16358   15516      4%      4%      4%   S          221376  fed main event
 14062   12756      1%      1%      1%   S           52140  sif_mgr
 32105    8618      0%      0%      0%   S             260  inotifywait
 31396   31393      0%      0%      0%   S           36516  python2.7
 31393   31271      0%      0%      0%   S            2744  rdope.sh
 31319       1      0%      0%      0%   S            2648  rotee
 31271       1      0%      0%      0%   S            3852  pman.sh
 29671       2      0%      0%      0%   S               0  kworker/u16:0
 29341   29329      0%      0%      0%   S            1780  sntp
 29329       1      0%      0%      0%   S            2788  stack_sntp.sh
.
```

```
.
.
```

The following is sample output from the **show processes cpu platform location switch 7 R0** command:

```
Device# show processes cpu platform location switch 7 R0

CPU utilization for five seconds:  3%, one minute:  3%, five minutes:  3%
Core 0: CPU utilization for five seconds:  1%, one minute:  5%, five minutes:  5%
Core 1: CPU utilization for five seconds:  1%, one minute: 11%, five minutes:  5%
Core 2: CPU utilization for five seconds: 22%, one minute:  7%, five minutes:  6%
Core 3: CPU utilization for five seconds:  5%, one minute:  6%, five minutes:  6%
Core 4: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 5: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 6: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 7: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  6%
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
--------------------------------------------------------------------------------
     1       0      0%      0%      0%  S             8044  systemd
     2       0      0%      0%      0%  S                0  kthreadd
.
.
.
```

# show processes cpu platform history

To display information about the CPU usage history of a system, use the **show processes cpu platform history** command.

**show processes cpu platform history** [ **1min** | **5min** | **5sec** | **60min** ] **location** **switch** { *switch-number* | **active** | **standby** } { **0** | **F0** | **FP active** | **R0** }

| | |
|---|---|
| **1min** | (Optional) Displays CPU utilization history with 1 minute intervals. |
| **5min** | (Optional) Displays CPU utilization history with 5 minute intervals. |
| **5sec** | (Optional) Displays CPU utilization history with 5 second intervals. |
| **60min** | (Optional) Displays CPU utilization history with 60 minute intervals. |
| **location** | Specifies the Field Replaceable Unit (FRU) location. |
| **switch** *switch-number* | Displays information about the switch. Enter the switch number. |
| **active** | Specifies the active instance of the device. |
| **standby** | Specifies the standby instance of the device. |
| **0** | Specifies the Shared Port Adapter (SPA) Interface Processor slot 0. |
| **F0** | Specifies the Embedded Service Processor (ESP) slot 0. |
| **FP active** | Specifies active instances on the Embedded Service Processor (ESP). |
| **R0** | Specifies the Route Processor (RP) slot 0. |

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Gibraltar 16.10.1 | This command was introduced. |

## Command Modes

Privileged EXEC (#)

### Examples:

The following is sample output from the **show processes cpu platform** command:

```
Device# show processes cpu platform
```

```
CPU utilization for five seconds:  1%, one minute:  3%, five minutes:  2%
Core 0: CPU utilization for five seconds:  2%, one minute:  2%, five minutes:  2%
Core 1: CPU utilization for five seconds:  2%, one minute:  1%, five minutes:  1%
Core 2: CPU utilization for five seconds:  3%, one minute:  1%, five minutes:  1%
Core 3: CPU utilization for five seconds:  2%, one minute:  5%, five minutes:  2%
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
--------------------------------------------------------------------------------
     1       0      0%      0%      0%  S             4876  systemd
     2       0      0%      0%      0%  S                0  kthreadd
     3       2      0%      0%      0%  S                0  ksoftirqd/0
     5       2      0%      0%      0%  S                0  kworker/0:0H
     7       2      0%      0%      0%  S                0  rcu_sched
     8       2      0%      0%      0%  S                0  rcu_bh
     9       2      0%      0%      0%  S                0  migration/0
    10       2      0%      0%      0%  S                0  watchdog/0
    11       2      0%      0%      0%  S                0  watchdog/1
    12       2      0%      0%      0%  S                0  migration/1
    13       2      0%      0%      0%  S                0  ksoftirqd/1
    15       2      0%      0%      0%  S                0  kworker/1:0H
    16       2      0%      0%      0%  S                0  watchdog/2
    17       2      0%      0%      0%  S                0  migration/2
    18       2      0%      0%      0%  S                0  ksoftirqd/2
    20       2      0%      0%      0%  S                0  kworker/2:0H
    21       2      0%      0%      0%  S                0  watchdog/3
    22       2      0%      0%      0%  S                0  migration/3
    23       2      0%      0%      0%  S                0  ksoftirqd/3
    24       2      0%      0%      0%  S                0  kworker/3:0
    25       2      0%      0%      0%  S                0  kworker/3:0H
    26       2      0%      0%      0%  S                0  kdevtmpfs
    27       2      0%      0%      0%  S                0  netns
    28       2      0%      0%      0%  S                0  perf
    29       2      0%      0%      0%  S                0  khungtaskd
    30       2      0%      0%      0%  S                0  writeback
    31       2      7%      8%      8%  S                0  ksmd
    32       2      0%      0%      0%  S                0  khugepaged
    33       2      0%      0%      0%  S                0  crypto
    34       2      0%      0%      0%  S                0  bioset
    35       2      0%      0%      0%  S                0  kblockd
    36       2      0%      0%      0%  S                0  ata_sff
    37       2      0%      0%      0%  S                0  rpciod
    63       2      0%      0%      0%  S                0  kswapd0
    64       2      0%      0%      0%  S                0  vmstat
    65       2      0%      0%      0%  S                0  fsnotify_mark
.
.
.
```

The following is sample output from the **show processes cpu platform history 5sec** command:

```
Device# show processes cpu platform history 5sec

5 seconds ago, CPU utilization: 0%
10 seconds ago, CPU utilization: 0%
15 seconds ago, CPU utilization: 0%
20 seconds ago, CPU utilization: 0%
25 seconds ago, CPU utilization: 0%
30 seconds ago, CPU utilization: 0%
35 seconds ago, CPU utilization: 0%
40 seconds ago, CPU utilization: 0%
45 seconds ago, CPU utilization: 0%
50 seconds ago, CPU utilization: 0%
55 seconds ago, CPU utilization: 0%
60 seconds ago, CPU utilization: 0%
65 seconds ago, CPU utilization: 0%
70 seconds ago, CPU utilization: 0%
```

```
75 seconds ago, CPU utilization: 0%
80 seconds ago, CPU utilization: 0%
85 seconds ago, CPU utilization: 0%
90 seconds ago, CPU utilization: 0%
95 seconds ago, CPU utilization: 0%
100 seconds ago, CPU utilization: 0%
105 seconds ago, CPU utilization: 0%
110 seconds ago, CPU utilization: 0%
115 seconds ago, CPU utilization: 0%
120 seconds ago, CPU utilization: 0%
125 seconds ago, CPU utilization: 0%
130 seconds ago, CPU utilization: 0%
135 seconds ago, CPU utilization: 0%
140 seconds ago, CPU utilization: 0%
145 seconds ago, CPU utilization: 1%
150 seconds ago, CPU utilization: 0%
155 seconds ago, CPU utilization: 0%
160 seconds ago, CPU utilization: 0%
165 seconds ago, CPU utilization: 0%
170 seconds ago, CPU utilization: 0%
175 seconds ago, CPU utilization: 0%
180 seconds ago, CPU utilization: 0%
185 seconds ago, CPU utilization: 0%
190 seconds ago, CPU utilization: 0%
195 seconds ago, CPU utilization: 0%
200 seconds ago, CPU utilization: 0%
205 seconds ago, CPU utilization: 0%
210 seconds ago, CPU utilization: 0%
215 seconds ago, CPU utilization: 0%
220 seconds ago, CPU utilization: 0%
225 seconds ago, CPU utilization: 0%
230 seconds ago, CPU utilization: 0%
235 seconds ago, CPU utilization: 0%
240 seconds ago, CPU utilization: 0%
245 seconds ago, CPU utilization: 0%
250 seconds ago, CPU utilization: 0%
.
.
.
```

# show processes cpu platform monitor

To displays information about the CPU utilization of the IOS-XE processes, use the **show processes cpu platform monitor** command in privileged EXEC mode.

**show processes cpu platform monitor location switch** {*switch-number* | **active** | **standby**} {**0** | **F0** | **R0**}

## Syntax Description

| | |
|---|---|
| **location** | Displays information about the Field Replaceable Unit (FRU) location. |
| **switch** | Specifies the switch. |
| *switch-number* | Switch number. |
| **active** | Specifies the active instance. |
| **standby** | Specifies the standby instance. |
| **0** | Specifies the shared port adapter (SPA) interface processor slot 0. |
| **F0** | Specifies the Embedded Service Processor (ESP) slot 0. |
| **R0** | Specifies the Route Processor (RP) slot 0. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

The output of the **show platform software process slot switch** and **show processes cpu platform monitor location** commands display the output of the Linux **top** command. The output of these commands display Free memory and Used memory as displayed by the Linux **top** command. The values displayed for the Free memory and Used memory by these commands do not match the values displayed by the output of other platform-memory related CLIs.

## Examples

The following is sample output from the **show processes cpu monitor location switch active R0** command:

```
Switch# show processes cpu platform monitor location switch active R0

top - 00:04:21 up 1 day, 11:22,  0 users,  load average: 0.42, 0.60, 0.78
Tasks: 312 total,   4 running, 308 sleeping,   0 stopped,   0 zombie
Cpu(s):  7.4%us,  3.3%sy,  0.0%ni, 89.2%id,  0.0%wa,  0.0%hi,  0.1%si,  0.0%st
Mem:   3976844k total,  3956928k used,    19916k free,   419312k buffers
Swap:        0k total,        0k used,        0k free,  1947036k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6294 root      20   0  3448 1368  912 R    9  0.0   0:00.07 top
17546 root      20   0 2044m 244m  79m S    7  6.3 187:02.07 fed main event
30276 root      20   0  171m  42m  33m S    7  1.1 125:15.54 repm
   16 root      20   0     0    0    0 S    5  0.0  22:07.92 rcuc/2
   21 root      20   0     0    0    0 R    5  0.0  22:13.24 rcuc/3
```

```
18662 root       20   0 1806m 678m 263m R    5 17.5 215:47.59 linux_iosd-imag
   11 root       20   0     0    0    0 S    4  0.0  21:37.41 rcuc/1
10333 root       20   0  6420 3916 1492 S    4  0.1   4:47.03 btrace_rotate.s
   10 root       20   0     0    0    0 S    2  0.0   0:58.13 rcuc/0
 6304 root       20   0   776   12    0 R    2  0.0   0:00.01 ls
17835 root       20   0  935m  74m  63m S    2  1.9  82:34.07 sif_mgr
    1 root       20   0  8440 4740 2184 S    0  0.1   0:09.52 systemd
    2 root       20   0     0    0    0 S    0  0.0   0:00.00 kthreadd
    3 root       20   0     0    0    0 S    0  0.0   0:02.86 ksoftirqd/0
    5 root        0 -20     0    0    0 S    0  0.0   0:00.00 kworker/0:0H
    7 root       RT   0     0    0    0 S    0  0.0   0:01.44 migration/0
```

## Related Commands

| Command | Description |
|---|---|
| **show platform software process slot switch** | Displays platform software process switch information. |

# show processes memory

To display the amount of memory used by each system process, use the **show processes memory** command in privileged EXEC mode.

**show processes memory** [{ *process-id* | **sorted** [{ **allocated** | **getbufs** | **holding** }] }]

**Syntax Description**

| | |
|---|---|
| *process-id* | (Optional) Process ID (PID) of a specific process. When you specify a process ID, only details for the specified process will be shown. |
| **sorted** | (Optional) Displays memory data sorted by the Allocated, Get Buffers, or Holding column. If the **sorted** keyword is used by itself, data is sorted by the Holding column by default. |
| **allocated** | (Optional) Displays memory data sorted by the Allocated column. |
| **getbufs** | (Optional) Displays memory data sorted by the Getbufs (Get Buffers) column. |
| **holding** | (Optional) Displays memory data sorted by the Holding column. This keyword is the default. |

**Command Modes**

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

The **show processes memory** command and the **show processes memory sorted** command displays a summary of total, used, and free memory, followed by a list of processes and their memory impact.

If the standard **show processes memory** *process-id* command is used, processes are sorted by their PID. If the **show processes memory sorted** command is used, the default sorting is by the Holding value.

**Note** Holding memory of a particular process can be allocated by other processes also, and so it can be greater than the allocated memory.

The following is sample output from the **show processes memory** command:

```
Device# show processes memory

Processor Pool Total:   25954228 Used:    8368640 Free:   17585588
 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0    8629528     689900    6751716          0          0 *Init*
   0   0      24048      12928      24048          0          0 *Sched*
   0   0        260        328         68     350080          0 *Dead*
   1   0          0          0      12928          0          0 Chunk Manager
   2   0        192        192       6928          0          0 Load Meter
   3   0     214664        304     227288          0          0 Exec
   4   0          0          0      12928          0          0 Check heaps
   5   0          0          0      12928          0          0 Pool Manager
   6   0        192        192      12928          0          0 Timers
   7   0        192        192      12928          0          0 Serial Backgroun
   8   0         192         192       12928           0           0 AAA high-capacit
   9   0           0           0       24928           0           0 Policy Manager
  10   0           0           0       12928           0           0 ARP Input
  11   0         192         192       12928           0           0 DDR Timers
  12   0           0           0       12928           0           0 Entity MIB API
  13   0           0           0       12928           0           0 MPLS HC Counter
  14   0           0           0       12928           0           0 SERIAL A'detect
.
.
.
  78   0           0           0       12992           0           0 DHCPD Timer
  79   0         160           0       13088           0           0 DHCPD Database
                                     8329440 Total
```

The table below describes the significant fields shown in the display.

*Table 14: show processes memory Field Descriptions*

| Field | Description |
|---|---|
| Processor Pool Total | Total amount of memory, in kilobytes (KB), held for the Processor memory pool. |
| Used | Total amount of used memory, in KB, in the Processor memory pool. |
| Free | Total amount of free memory, in KB, in the Processor memory pool. |
| PID | Process ID. |
| TTY | Terminal that controls the process. |
| Allocated | Bytes of memory allocated by the process. |
| Freed | Bytes of memory freed by the process, regardless of who originally allocated it. |
| Holding | Amount of memory, in KB, currently allocated to the process. This includes memory allocated by the process and assigned to the process. |
| Getbufs | Number of times the process has requested a packet buffer. |
| Retbufs | Number of times the process has relinquished a packet buffer. |
| Process | Process name. |
| *Init* | System initialization process. |
| *Sched* | The scheduler process. |
| *Dead* | Processes as a group that are now dead. |
| <value> Total | Total amount of memory, in KB, held by all processes (sum of the “Holding” column). |

The following is sample output from the **show processes memory** command when the **sorted** keyword is used. In this case, the output is sorted by the Holding column, from largest to smallest.

```
Device# show processes memory sorted

Processor Pool Total:   25954228 Used:    8371280 Free:   17582948
 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0    8629528     689900    6751716          0          0 *Init*
   3   0     217304        304     229928          0          0 Exec
  53   0     109248        192      96064          0          0 DHCPD Receive
  56   0          0          0      32928          0          0 COPS
  19   0      39048          0      25192          0          0 Net Background
  42   0          0          0      24960          0          0 L2X Data Daemon
  58   0        192        192      24928          0          0 X.25 Background
  43   0        192        192      24928          0          0 PPP IP Route
  49   0          0          0      24928          0          0 TCP Protocols
  48   0          0          0      24928          0          0 TCP Timer
  17   0        192        192      24928          0          0 XML Proxy Client
   9   0          0          0      24928          0          0 Policy Manager
  40   0          0          0      24928          0          0 L2X SSS manager
  29   0          0          0      24928          0          0 IP Input
  44   0        192        192      24928          0          0 PPP IPCP
  32   0        192        192      24928          0          0 PPP Hooks
  34   0          0          0      24928          0          0 SSS Manager
  41   0        192        192      24928          0          0 L2TP mgmt daemon
  16   0        192        192      24928          0          0 Dialer event
  35   0          0          0      24928          0          0 SSS Test Client
 --More--
```

The following is sample output from the **show processes memory** command when a process ID (*process-id*) is specified:

```
Device# show processes memory 1

Process ID: 1
Process Name: Chunk Manager
Total Memory Held: 8428 bytes
Processor memory holding = 8428 bytes
pc = 0x60790654, size =      6044, count =    1
pc = 0x607A5084, size =      1544, count =    1
pc = 0x6076DBC4, size =       652, count =    1
pc = 0x6076FF18, size =       188, count =    1
I/O memory holding = 0 bytes

Device# show processes memory 2

Process ID: 2
Process Name: Load Meter
Total Memory Held: 3884 bytes
Processor memory holding = 3884 bytes
pc = 0x60790654, size =      3044, count =    1
pc = 0x6076DBC4, size =       652, count =    1
pc = 0x6076FF18, size =       188, count =    1
I/O memory holding = 0 bytes
```

## Related Commands

| Command | Description |
|---|---|
| **show memory** | Displays statistics about memory, including memory-free pool statistics. |
| **show processes** | Displays information about the active processes. |

# show processes memory platform

To display memory usage for each Cisco IOS XE process, use the **show processes memory platform** command in privileged EXEC mode.

**show processes memory platform** [ [ **detailed** { **name** *process-name* | **process-id** *process-ID* } [ **location** | **maps** [ **location** ] | **smaps** [ **location** ] ] | **location** | **sorted** [ **location** ] ] **switch** { *switch-number* | **active** | **standby** } { **0** | **F0** | **R0** } | **accounting** ]

## Syntax Description

| | |
|---|---|
| **accounting** | (Optional) Displays the top memory allocators for each Cisco IOS XE process. |
| **detailed** | (Optional) Displays detailed memory information for a specified Cisco IOS XE process. |
| **name** *process-name* | (Optional) Displays the Cisco IOS XE process name. Enter the process name. |
| **process-id** *process-ID* | (Optional) Displayss the Cisco IOS XE process ID. Enter the process ID. |
| **location** | (Optional) Displays information about the Field Replaceable Unit (FRU) location. |
| **maps** | (Optional) Displays memory maps of a process. |
| **smaps** | (Optional) Displays static memory maps of a process. |
| **sorted** | (Optional) Displays the sorted output based on the Resident Set Size (RSS) memory used by Cisco IOS XE process. |
| **switch** *switch-number* | Displays information about the device. |
| **active** | Displays information about the active instance of the device. |
| **standby** | Displays information about the standby instance of the device. |
| **0** | Displays information about Shared Port Adapter (SPA)-Inter-Processor slot 0. |
| **F0** | Displays information about Embedded Service Processor (ESP) slot 0. |
| **R0** | Displays information about Route Processor (RP) slot 0. |

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |
| Cisco IOS XE Gibraltar 16.10.1 | This command was modified. The keyword **accounting** was added.<br>The **Total** column was deleted from the output. |

## Examples

The following is a sample output from the **show processes memory platform** command:

```
device# show processes memory platform

System memory: 3976852K total, 2761580K used, 1215272K free,
Lowest: 1215272K
   Pid    Text      Data   Stack   Dynamic       RSS                  Name
----------------------------------------------------------------------------
     1    1246      4400     132      1308      4400               systemd
    96     233      2796     132       132      2796       systemd-journal
   105     284      1796     132       176      1796         systemd-udevd
   707      52      2660     132       172      2660            in.telnetd
   744     968      3264     132      1700      3264             brelay.sh
   835      52      2660     132       172      2660            in.telnetd
   863     968      3264     132      1700      3264             brelay.sh
   928     968      3996     132      2312      3996          reflector.sh
   933     968      3976     132      2312      3976           droputil.sh
   934     968      2140     132       528      2140                oom.sh
   936     173       936     132       132       936                xinetd
   945     968      1472     132       132      1472           libvirtd.sh
   947     592     43164     132      3096     43164                  repm
   954      45       932     132       132       932               rpcbind
   986     482      3476     132       132      3476              libvirtd
   988      66       940     132       132       940             rpc.statd
   993     968       928     132       132       928       boothelper_evt.
  1017      21       640     132       132       640           inotifywait
  1089     102      1200     132       132      1200            rpc.mountd
  1328       9      2940     132       148      2940                 rotee
  1353      39       532     132       132       532                 sleep
!
!
!
```

The following is a sample output from the **show processes memory platform accounting** command:

```
device# show processes memory platform accounting
Hourly Stats

  process                 callsite_ID(bytes)  max_diff_bytes   callsite_ID(calls)
max_diff_calls   tracekey                                  timestamp(UTC)

__________________________________________________________________________________________

  smand_rp_0              3624155137          172389           3624155138          50
          1#a3e0e4361082c702e5bf1afbd90e6313        2018-09-04 14:23
  linux_iosd-imag_rp_0    3626295305          49188            3624155138          12
          1#545420bd869d25eb5ab826182ee5d9ce        2018-09-04 12:03
  btman_rp_0              3624737792          17080            2953915394          64
          1#d6888bd9564a3c4fcf049c31ba07a036        2018-09-04 22:29
  fman_fp_image_fp_0      3624059905          16960           4027402242          298
          1#921ba4d9df5b0a6e946a3b270bd6592d        2018-09-04 22:55
  fed_main_event_fp_0     3626295305          16396           4027402242          32
          1#27083f7bf3985d892505806cae2bfb0d        2018-09-04 12:03
  dbm_rp_0                3626295305          16396           4027402242          3
          1#2b878f802bd7703c5298d37e7a4e8ac3        2018-09-04 12:02
  tamd_proc_rp_0          3895208962          12632           3624667171          7
          1#5b0ed8f88ef5f873abcaf8a744037a44        2018-09-04 18:47
  btman_fp_0              3624233985          12288           3624737792          9
          1#d6888bd9564a3c4fcf049c31ba07a036        2018-09-04 15:23
  sif_mgr_rp_0            3624059907          8216            4027402242          4
          1#de2a951a8a7bae83ca2c04c56810eb72        2018-09-04 14:21
  python2.7_fp_0          2954560513          8000            2954560513          1
                                                    2018-09-04 12:16
  nginx_rp_0              3357041665          4608            4027402242          4
          1#32e56bb09e0509c5fa5ac32093631206        2018-09-04 16:18
  rotee_FRU_SLOT_NUM      3624667169          4097            3624667169          1
          1#ff68e5150a698cd59fa259828614995b        2018-09-04 10:43
  hman_rp_0               3893617664          1488            3893617664          1
          1#1c4aadada30083c5d6f66dc8ca8cd4cb        2018-09-04 10:42
  tams_proc_rp_0          3895096320          1024            3895096320          1
          1#a36a3afa9884c8dc4d40af1e80cacd26        2018-09-04 10:42
  stack_mgr_rp_0          4027402242          904             4027402242          4
          1#ca902eab11a18ab056b16554f49871e8        2018-09-04 14:21
  sessmgrd_rp_0           3491618816          848             3624155138          8
          1#720239fc8bddcabc059768c55a1640ed        2018-09-04 14:32
  psd_rp_0                4027402242          696             4027402242          4
          1#98cf04e0ddd78c2400b3ca3b5f298594        2018-09-04 14:21
  lman_rp_0               4027402242          592             4027402242          4
          1#dc8ed9e428d36477a617d56c51d5caf2        2018-09-04 14:21
  bt_logger_rp_0          4027402242          592             4027402242          4
          1#ba882be1ed783e72575e97cc0908e0e8        2018-09-04 14:21
  repm_rp_0               4027402242          592             4027402242          4
          1#ae461a05430efa767427f2ab40aba372        2018-09-04 14:21
  fman_rp_rp_0            4027402242          592             4027402242          3
          1#09def9cc1390911be9e3a7a9c89f4cf7        2018-09-04 12:16
  epc_ws_liaison_fp_0     4027402242          592             4027402242          4
          1#41451626dcce9d1478b22e2ebbbdcf54        2018-09-04 14:21
  cli_agent_rp_0          4027402242          592             4027402242          4
          1#92d3882919daf3a9e210807c61de0552        2018-09-04 14:21
  cmm_rp_0                4027402242          592             4027402242          4
          1#15ed1d79e96874b1e0621c42c3de6166        2018-09-04 14:21
  tms_rp_0                4027402242          352             4027402242          4
          1#5c6efe2e21f15aa16318576d3ec9153c        2018-09-04 12:03
  plogd_rp_0              4027402242          48              4027402242          1
          1#2d7f2ef57206f4fa763d7f2f5400bf1b        2018-09-04 10:43
  cmand_rp_0              3624155137          17              3624155137          1
          1#f1f41f61c44d73014023db5d8a46ecf5        2018-09-04 10:42
!
!
!
```

The following is a sample output from the **show processes memory platform sorted** command:

```
device# show processes memory platform sorted
System memory: 3976852K total, 2762884K used, 1213968K free,
Lowest: 1213968K

   Pid      Text      Data   Stack   Dynamic       RSS                Name
--------------------------------------------------------------------------
  7885    149848     684864    136        80    684864     linux_iosd-imag
  9655      3787     264964    136     18004    264964                 wcm
  17261     324      248588    132     103908    248588      fed main event
  4268      391      102084    136       5596    102084           cli_agent
  4856      357       93388    132       3680     93388                 dbm
17067     1087       77912    136       1796     77912        platform_mgr
!
!
!
```

The following is sample output from the **show processes memory platform sorted location switch active R0** command:

```
device# show processes memory platform sorted location switch active R0
System memory: 3976852K total, 2762884K used, 1213968K free,
Lowest: 1213968K

   Pid      Text      Data   Stack    Dynamic       RSS                Name
---------------------------------------------------------------------------
  7885    149848     684864    136         80    684864     linux_iosd-imag
  9655      3787     264964    136      18004    264964                 wcm
  17261     324      248588    132     103908    248588      fed main event
  4268      391      102084    136       5596    102084           cli_agent
  4856      357       93388    132       3680     93388                 dbm
17067     1087       77912    136       1796     77912        platform_mgr
!
!
!
```

# show processes platform

To display information about the IOS-XE processes running on a platform, use the **show processes platform** command in privileged EXEC mode.

**show processes platform** [ **detailed name** *process-name* ] [ **location switch** { *switch-number* | **active** | **standby** } { **0** | **F0** | **FP active** | **R0** } ]

| | |
|---|---|
| **detailed** | (Optional) Displays detailed information of the specified IOS-XE process. |
| **name** *process-name* | (Optional) Specifies the process name. |
| **location** | (Optional) Specifies the Field Replaceable Unit (FRU) location. |
| **switch** *switch-number* | (Optional) Displays information about the switch. |
| **active** | (Optional) Specifies the active instance of the device. |
| **standby** | (Optional) Specifies standby instance of the device. |
| **0** | Specifies the Shared Port Adapter (SPA) Interface Processor slot 0. |
| **F0** | Specifies the Embedded Service Processor (ESP) slot 0. |
| **FP active** | Specifies the active instance in the Embedded Service Processor (ESP). |
| **R0** | Specifies the Route Processor (RP) slot 0. |

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Gibraltar 16.10.1 | This command was introduced. |

## Command Modes

Privileged EXEC(#)

### Examples:

The following is sample output from the **show processes platform** command:

```
Device# show processes platform

CPU utilization for five seconds: 1%, one minute: 2%, five minutes: 1%
   Pid    PPid  Status        Size  Name
--------------------------------------------------------
     1       0  S             4876  systemd
     2       0  S                0  kthreadd
     3       2  S                0  ksoftirqd/0
     5       2  S                0  kworker/0:0H
     7       2  S                0  rcu_sched
     8       2  S                0  rcu_bh
     9       2  S                0  migration/0
    10       2  S                0  watchdog/0
    11       2  S                0  watchdog/1
    12       2  S                0  migration/1
   13       2  S                0  ksoftirqd/1
   15       2  S                0  kworker/1:0H
   16       2  S                0  watchdog/2
   17       2  S                0  migration/2
   18       2  S                0  ksoftirqd/2
   20       2  S                0  kworker/2:0H
   21       2  S                0  watchdog/3
   22       2  S                0  migration/3
   23       2  S                0  ksoftirqd/3
   24       2  S                0  kworker/3:0
   25       2  S                0  kworker/3:0H
   26       2  S                0  kdevtmpfs
   27       2  S                0  netns
   28       2  S                0  perf
   29       2  S                0  khungtaskd
   30       2  S                0  writeback
   31       2  S                0  ksmd
   32       2  S                0  khugepaged
   33       2  S                0  crypto
   34       2  S                0  bioset
   35       2  S                0  kblockd
   36       2  S                0  ata_sff
   37       2  S                0  rpciod
   63       2  S                0  kswapd0
   64       2  S                0  vmstat
   65       2  S                0  fsnotify_mark
   66       2  S                0  nfsiod
   74       2  S                0  bioset
   75       2  S                0  bioset
   76       2  S                0  bioset
   77       2  S                0  bioset
   78       2  S                0  bioset
   79       2  S                0  bioset
   80       2  S                0  bioset
   81       2  S                0  bioset
   82       2  S                0  bioset
   83       2  S                0  bioset
   84       2  S                0  bioset
   85       2  S                0  bioset
   86       2  S                0  bioset
   87       2  S                0  bioset
   88       2  S                0  bioset
   89       2  S                0  bioset
   90       2  S                0  bioset
   91       2  S                0  bioset
   92       2  S                0  bioset
   93       2  S                0  bioset
   94       2  S                0  bioset
   95       2  S                0  bioset
   96       2  S                0  bioset
   97       2  S                0  bioset
  100       2  S                0  ipv6_addrconf
  102       2  S                0  deferwq
```

The table below describes the significant fields shown in the displays.

***Table 15: show processes platform Field Descriptions***

| Field | Description |
|---|---|
| Pid | Displays the process ID. |

| **Field** | **Description** |
|---|---|
| PPid | Displays the process ID of the parent process. |
| Status | Displays the process status in human readable form. |
| Size | Displays the Resident Set Size (in kilobytes (KB)) that shows how much memory is allocated to that process in the RAM. |
| Name | Displays the command name associated with the process. Different threads in the same process may have different command values. |

# show shell

To display shell information, use the **show shell** command in user EXEC mode.

**show shell** [{**enviornment** | **functions** [{**brief** *shell_function*}] | **triggers**}]

## Syntax Description

| | |
|---|---|
| **environment** | (Optional) Displays shell environment inform |
| **functions** [**brief** \|*shell_function* ] | (Optional) Displays macro information.<br>• **brief**—Names of the shell functions.<br>• *shell_function*—Name of a shell functio |
| **triggers** | (Optional) Displays event trigger information. |

## Command Modes

User EXEC (>)

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use this command to display the shell information for the switch.

### Example

This example shows how to use the **show shell** *triggers* command to view the event triggers in the switch software:

```
Device# term shell
Device# show shell triggers
User defined triggers
---------------------
Built-in triggers
-----------------
Trigger Id: CISCO_CUSTOM_EVENT
Trigger description: Custom macroevent to apply user defined configuration
Trigger environment: User can define the macro
Trigger mapping function: CISCO_CUSTOM_AUTOSMARTPORT

Trigger Id: CISCO_DMP_EVENT
Trigger description: Digital media-player device event to apply port configuration
Trigger environment: Parameters that can be set in the shell - $ACCESS_VLAN=(1)
 The value in the parenthesis is a default value
Trigger mapping function: CISCO_DMP_AUTO_SMARTPORT

Trigger Id: CISCO_IPVSC_EVENT
Trigger description: IP-camera device event to apply port configuration
Trigger environment: Parameters that can be set in the shell - $ACCESS_VLAN=(1)
 The value in parenthesis is a default value
Trigger mapping function: CISCO_IP_CAMERA_AUTO_SMARTPORT
Trigger Id: CISCO_LAST_RESORT_EVENT
Trigger description: Last resortevent to apply port configuration
Trigger environment: Parameters that can be set in the shell - $ACCESS_VLAN=(1)
 The value in the parenthesis is a default value
Trigger mapping function: CISCO_LAST_RESORT_SMARTPORT

Trigger Id: CISCO_PHONE_EVENT
Trigger description: IP-phone device event to apply port configuration
Trigger environment: Parameters that can be set in the shell - $ACCESS_VLAN=(1)
and $VOICE_VLAN=(2), The value in the parenthesis is a default value
Trigger mapping function: CISCO_PHONE_AUTO_SMARTPORT

Trigger Id: CISCO_ROUTER_EVENT
Trigger description: Router device event to apply port configuration
Trigger environment: Parameters that can be set in the shell - $NATIVE_VLAN=(1)
 The value in the parenthesis is a default value
Trigger mapping function: CISCO_ROUTER_AUTO_SMARTPORT

Trigger Id: CISCO_SWITCH_ETHERCHANNEL_CONFIG
Trigger description: etherchannel parameter
Trigger environment: $INTERFACE_LIST=(),$PORT-CHANNEL_ID=(),
                               $EC_MODE=(),$EC_PROTOCOLTYPE=(),
                               PORT-CHANNEL_TYPE=()
Trigger mapping function: CISCO_ETHERCHANNEL_AUTOSMARTPORT

Trigger Id: CISCO_SWITCH_EVENT
Trigger description: Switch device event to apply port configuration
Trigger environment: Parameters that can be set in the shell - $NATIVE_VLAN=(1)
 The value in the parenthesis is a default value
Trigger mapping function: CISCO_SWITCH_AUTO_SMARTPORT

Trigger Id: CISCO_WIRELESS_AP_EVENT
Trigger description: Autonomous ap device event to apply port configuration
Trigger environment: Parameters that can be set in the shell - $NATIVE_VLAN=(1)
 The value in the parenthesis is a default value
Trigger mapping function: CISCO_AP_AUTO_SMARTPORT

Trigger Id: CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT
Trigger description: Lightweight-ap device event to apply port configuration
Trigger environment: Parameters that can be set in the shell - $ACCESS_VLAN=(1)
  The value in the parenthesis is a default value
Trigger mapping function: CISCO_LWAP_AUTO_SMARTPORT

Trigger Id: word
Trigger description: word
Trigger environment:
Trigger mapping function:
```

This example shows how to use the **show shell functions** command to view the built-in macros in the switch software:

```
Device# show shell functions
#User defined functions:

#Built-in functions:
function CISCO_AP_AUTO_SMARTPORT () {
    if [[ $LINKUP == YES ]]; then
        conf t
            interface  $INTERFACE
                macro description $TRIGGER
                switchport trunk encapsulation dot1q
                switchport trunk native vlan $NATIVE_VLAN
                switchport trunk allowed vlan ALL
                switchport mode trunk
                switchport nonegotiate
                auto qos voip trust
                mls qos trust cos
                if [[ $LIMIT == 0 ]]; then
                  default srr-queue bandwidth limit
                else
                  srr-queue bandwidth limit $LIMIT
                fi
                if [[ $SW_POE == YES ]]; then
                   if [[ $AP125X == AP125X ]]; then
                      macro description AP125X
                      macro auto port sticky
                      power inline port maximum 20000
                   fi
                fi
             exit
        end
    fi
    if [[ $LINKUP == NO ]]; then
        conf t
            interface  $INTERFACE
                no macro description
                no switchport nonegotiate
                no switchport trunk native vlan $NATIVE_VLAN
                no switchport trunk allowed vlan ALL
                no auto qos voip trust
                no mls qos trust cos
                default srr-queue bandwidth limit
                if [[ $AUTH_ENABLED == NO ]]; then
                  no switchport mode
                  no switchport trunk encapsulation
                fi
                if [[ $STICKY == YES ]]; then
                   if [[ $SW_POE == YES ]]; then
                      if [[ $AP125X == AP125X ]]; then
                         no macro auto port sticky
                         no power inline port maximum
                      fi
                   fi
                fi
            exit
        end
    fi
}
<output truncated>
```

# show system mtu

To display the global maximum transmission unit (MTU) or maximum packet size set for the switch, use the **show system mtu** command in privileged EXEC mode.

**show system mtu**

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

**Command Default** None

**Command Modes** Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines** For information about the MTU values and the stack configurations that affect the MTU values, see the **system mtu** command.

**Examples**

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

```
Device# show system mtu
Global Ethernet MTU is 1500 bytes.
```

# show tech-support

To automatically run **show** commands that display system information, use the **show tech-support** command in the privilege EXEC mode.

**show tech-support**
[**cef** | **cft** | **eigrp** | **evc** | **fnf** |  | **ipc** | **ipmulticast** | **ipsec** | **mfib** | **nat** | **nbar** | **onep** | **ospf** | **page** | **password** | **rsvp** | **subscriber** | **vrrp** | **wccp**

## Syntax Description

| | |
|---|---|
| **cef** | (Optional) Displays CEF related information. |
| **cft** | (Optional) Displays CFT related information. |
| **eigrp** | (Optional) Displays EIGRP related information. |
| **evc** | (Optional) Displays EVC related information. |
| **fnf** | (Optional) Displays flexible netflow related information. |
| **ipc** | (Optional) Displays IPC related information. |
| **ipmulticast** | (Optional) Displays IP multicast related information. |
| **ipsec** | (Optional) Displays IPSEC related information. |
| isis | (Optional) Displays CLNS and ISIS related information. |
| license | (Optional) Displays license related information. |
| lisp | (Optional) Displays Locator/ID Separation Protocol related information. |
| memory | (Optional) Displays Memory related information. |
| **mfib** | (Optional) Displays MFIB related information. |
| msrp | (Optional) Displays MSRP related information. |
| mvrp | (Optional) Displays MVRP related information. |
| **nat** | (Optional) Displays NAT related information. |
| **onep** | (Optional) Displays ONEP related information. |
| **ospf** | (Optional) Displays OSPF related information. |
| **page** | (Optional) Displays the command output on a single page at a time. Use the Return key to display the next line of output or use the space bar to display the next page of information. If not used, the output scrolls (that is, it does not stop for page breaks).<br>Press the **Ctrl-C** keys to stop the command output. |
| **password** | (Optional) Leaves passwords and other security information in the output. If not used, passwords and other security-sensitive information in the output are replaced with the label "<removed>". |

| | |
|---|---|
| performance-monitor | (Optional) Displays Performance Monitor related information. |
| pki | (Optional) Displays PKI related information. |
| platform | (Optional) Displays Platform related information. |
| qos | (Optional) Displays QoS related information. |
| **subscriber** | (Optional) Displays subscriber related information. |
| switch-report | (Optional) Archives switch report. |
| **vrrp** | (Optional) Displays VRRP related information. |
| **wccp** | (Optional) Displays WCCP related information. |

**Command Modes**

Privileged EXEC (#)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was implemented. |

**Usage Guidelines**

The output from the **show tech-support** command is very long. To better manage this output, you can redirect the output to a file (for example, **show tech-support** > *filename* ) in the local writable storage file system or the remote file system. Redirecting the output to a file also makes sending the output to your Cisco Technical Assistance Center (TAC) representative easier.

You can use one of the following redirection methods:

- > *filename* - Redirects the output to a file.
- >> *filename* - Redirects the output to a file in append mode.

# show tech-support bgp

To automatically run show commands that display BGP related system information, use the **show tech-support bgp** command in the privileged EXEC mode.

**show tech-support bgp** [**address-family** {**all** | **ipv4** [**flowspec** | **multicast** | **unicast** | [**mdt** | **mvpn**] {**all** | **vrf** *vrf-instance-name*} ] |**ipv6** [**flowspec** | **multicast** | **mvpn** {**all** | **vrf** *vrf-instance-name*} | **unicast**] | **l2vpn** [**evpn** | **vpls**] | **link-state** [**link-state**] | [**nsap** | **rtfilter**] [**unicast**] | [**vpnv4** | **vpnv6**] [**flowspec** | **multicast** | **unicast**] {**all** | **vrf** *vrf-instance-name*}}] [**detail**]

**Syntax Description**

| | |
|---|---|
| **address-family** | (Optional) Displays the output for a specified address family. |
| **address-family all** | (Optional) Displays the output for all address families. |
| **ipv4** | (Optional) Displays the output for IPv4 address family. |
| **ipv6** | (Optional) Displays the output for IPv6 address family. |
| **l2vpn** | (Optional) Displays the output for L2VPN address family. |
| **link-state** | (Optional) Displays the output for Link State address family. |
| **nsap** | (Optional) Displays the output for NSAP address family. |
| **rtfilter** | (Optional) Displays the output for RT Filter address family. |
| **vpnv4** | (Optional) Displays the output for VPNv4 address family. |
| **vpnv6** | (Optional) Displays the output for VPNv6 address family. |
| **flowspec** | (Optional) Displays the flowspec related information for an address family. |
| **multicast** | (Optional) Displays the multicast related information for an address family. |
| **unicast** | (Optional) Displays the unicast related information for an address family. |
| **mdt** | (Optional) Displays the Multicast Distribution Tree (MDT) related information for an address family. |

| | |
|---|---|
| **mvpn** | (Optional) Displays the Multicast VPN (MVPN) related information for an address family. |
| **vrf** | Displays the information for a VPN Routing/Forwarding instance. |
| **evpn** | (Optional) Displays the Ethernet VPN (EVPN) related information for an address family. |
| **vpls** | (Optional) Displays the Virtual Private LAN Services (VPLS) related information for an address family. |
| *vrf-instance-name* | Specifies the name of the VPN Routing/Forwarding instance. |
| **all** | Displays the information about all VPN NLRIs. |
| **detail** | (Optional) Displays the detailed routes information. |

## Command Modes

User EXEC (>)

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| | This command was introduced. |

## Usage Guidelines

The **show tech-support bgp** command is used to display the outputs of various BGP show commands and log them to the show-tech file. The output from the **show tech-support bgp** command is very long. To better manage this output, you can redirect the output to a file (for example, **show tech-support** > **filename** ) in the local writable storage file system or the remote file system. Redirecting the output to a file also makes sending the output to your Cisco Technical Assistance Center (TAC) representative easier.

You can use one of the following redirection methods:

- > filename - Redirects the output to a file.
- >> filename - Redirects the output to a file in append mode.

The following **show** commands run automatically when the **show tech-support bgp** command is used:

- **show clock**
- **show version**
- **show running-config**
- **show process cpu sorted**
- **show process cpu history**
- **show process memory sorted**

The following **show** commands for a specific address family run automnatically when the **show tech-support bgp address-family***address-family-name address-family-modifier* command is used:

- **show bgp** *address-family-name address-family-modifier* **summary**
- **show bgp** *address-family-name address-family-modifier* **detail**
- **show bgp** *address-family-name address-family-modifier* **internal**
- **show bgp** *address-family-name address-family-modifier* **neighbors**
- **show bgp** *address-family-name address-family-modifier* **update-group**
- **show bgp** *address-family-name address-family-modifier* **replication**
- **show bgp** *address-family-name address-family-modifier* **community**
- **show bgp** *address-family-name address-family-modifier* **dampening dampened-paths**
- **show bgp** *address-family-name address-family-modifier* **dampening flap-statistics**
- **show bgp** *address-family-name address-family-modifier* **dampening parameters**
- **show bgp** *address-family-name address-family-modifier* **injected-paths**
- **show bgp** *address-family-name address-family-modifier* **cluster-ids**
- **show bgp** *address-family-name address-family-modifier* **cluster-ids internal**
- **show bgp** *address-family-name address-family-modifier* **peer-group**
- **show bgp** *address-family-name address-family-modifier* **pending-prefixes**
- **show bgp** *address-family-name address-family-modifier* **rib-failure**

In addition to the above commands, the following segment routing specific **show** commands also run when the **show tech-support bgp** command is used:

- **show bgp all binding-sid**
- **show segment-routing client**
- **show segment-routing mpls state**
- **show segment-routing mpls gb**
- **show segment-routing mpls connected-prefix-sid-map protocol ipv4**
- **show segment-routing mpls connected-prefix-sid-map protocol backup ipv4**
- **show mpls traffic-eng tunnel auto-tunnel client bgp**

# show tech-support diagnostic

To display diagnostic information for technical support, use the **show tech-support diagnostic** command in privileged EXEC mode.

**show tech-support diagnostic**

## Syntax Description

This command has no arguments or keywords.

## Command Modes

Privileged EXEC (#)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Gibraltar 16.10.1 | This command was introduced. |

## Usage Guidelines

The output of this command is very long. To better manage this output, you can redirect the output to a file (for example, **show tech-support diagnostic** > **flash:***filename* ) in the local writable storage file system or remote file system.

**Note** For devices that support stacking, this command is executed on every switch that is up. For devices that do not support stacking, this command is executed only on the active switch.

The output of this command displays the output of the following commands:

- **show clock**
- **show version**
- **show running-config**
- **show inventory**
- **show diagnostic bootup level**
- **show diagnostic status**
- **show diagnostic content switch all**
- **show diagnostic result switch all detail**
- **show diagnostic schedule switch all**
- **show diagnostic post**
- **show diagnostic description switch [switch number] test all**
- **show logging onboard switch [switch number] clilog detail**
- **show logging onboard switch [switch number] counter detail**
- **show logging onboard switch [switch number] environment detail**
- **show logging onboard switch [switch number] message detail**

- **show logging onboard switch [switch number] poe detail**
- **show logging onboard switch [switch number] status**
- **show logging onboard switch [switch number] temperature detail**
- **show logging onboard switch [switch number] uptime detail**
- **show logging onboard switch [switch number] voltage detail**

# speed

To specify the speed of a port, use the **speed** command in interface configuration mode. To return to the default value, use the **no** form of this command.

**Note** Available configuration options depend on the switch model and transceiver module installed. Options include 10, 100, 1000, 2500, 5000, 10000, 25000, 40000, 100000

**speed** {**10** | **100** | **1000** | **2500** | **5000** | **auto** [{**10** | **100** | **1000** | **2500** | **5000**}] | **nonegotiate**}
**no speed**

## Syntax Description

| | |
|---|---|
| **10** | Specifies that the port runs at 10 Mbps. |
| **100** | Specifies that the port runs at 100 Mbps. |
| **1000** | Specifies that the port runs at 1000 Mbps. This option is valid and visible only on 10/100/1000 Mb/s ports. |
| **2500** | Specifies that the port runs at 2500 Mbps. This option is valid and visible only on multi-Gigabit-supported Ethernet ports. |
| **5000** | Specifies that the port runs at 5000 Mbps. This option is valid and visible only on multi-Gigabit-supported Ethernet ports. |
| **auto** | Detects the speed at which the port should run, automatically, based on the port at the other end of the link. If you use the **10**, **100**, **1000**, **2500**, or **5000** keyword with the **auto** keyword, the port autonegotiates only at the specified speeds. |
| **nonegotiate** | Disables autonegotiation, and the port runs at 1000 Mbps. |

## Command Default

The default is **auto**.

## Command Modes

Interface configuration (config-if)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

You cannot configure speed on 10-Gigabit Ethernet ports.

Except for the 1000BASE-T small form-factor pluggable (SFP) modules, you can configure the speed to not negotiate (**nonegotiate**) when an SFP module port is connected to a device that does not support autonegotiation.

The keywords, **2500** and **5000** are visible only on multi-Gigabit (m-Gig) Ethernet supporting devices.

If the speed is set to **auto**, the switch negotiates with the device at the other end of the link for the speed setting, and then forces the speed setting to the negotiated value. The duplex setting remains configured on each end of the link, which might result in a duplex setting mismatch.

If both ends of the line support autonegotiation, we highly recommend the default autonegotiation settings. If one interface supports autonegotiation and the other end does not, use the auto setting on the supported side, but set the duplex and speed on the other side.

**Caution** Changing the interface speed and duplex mode configuration might shut down and re-enable the interface during the reconfiguration.

For guidelines on setting the switch speed and duplex parameters, see the "Configuring Interface Characteristics" chapter in the software configuration guide for this release.

Verify your settings using the **show interfaces** privileged EXEC command.

## Examples

The following example shows how to set speed on a port to 100 Mbps:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# speed 100
```

The following example shows how to set a port to autonegotiate at only 10 Mbps:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# speed auto 10
```

The following example shows how to set a port to autonegotiate at only 10 or 100 Mbps:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# speed auto 10 100
```

# start (coap-proxy configuration)

To start CoAP on the switch, use the **start** command in coap-proxy configuration mode.

**start**

**Command Modes**

coap-proxy configuration (config-coap-proxy)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

To access coap-proxy configuration mode, enter the **coap proxy** command in global configuration mode.

### Example

This example shows how to start CoAP on the switch.

```
Device(config)# coap proxy
Device(config-coap-proxy)# start
```

# stop (coap-proxy configuration)

To stop CoAP on the switch, use the **stop** command in coap-proxy configuration mode.

**stop**

**Command Modes**

coap-proxy configuration (config-coap-proxy)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

To access coap-proxy configuration mode, enter the **coap proxy** command in global configuration mode.

### Example

This example shows how to stop CoAP on the switch.

```
Device(config)# coap proxy
Device(config-coap-proxy)# stop
```

# switchport block

To prevent unknown multicast or unicast packets from being forwarded, use the **switchport block** command in interface configuration mode. To allow forwarding unknown multicast or unicast packets, use the **no** form of this command.

**switchport block** {**multicast** | **unicast**}
**no switchport block** {**multicast** | **unicast**}

## Syntax Description

| | |
|---|---|
| **multicast** | Specifies that unknown multicast traffic should be blocked.<br>**Note** Only pure Layer 2 multicast traffic is blocked. Multicast packets that contain IPv4 or IPv6 information in the header are not blocked. |
| **unicast** | Specifies that unknown unicast traffic should be blocked. |

## Command Default

Unknown multicast and unicast traffic is not blocked.

## Command Modes

Interface configuration (config-if)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

By default, all traffic with unknown MAC addresses is sent to all ports. You can block unknown multicast or unicast traffic on protected or nonprotected ports. If unknown multicast or unicast traffic is not blocked on a protected port, there could be security issues.

With multicast traffic, the port blocking feature blocks only pure Layer 2 packets. Multicast packets that contain IPv4 or IPv6 information in the header are not blocked.

Blocking unknown multicast or unicast traffic is not automatically enabled on protected ports; you must explicitly configure it.

For more information about blocking packets, see the software configuration guide for this release.

This example shows how to block unknown unicast traffic on an interface:

```
Device(config-if)# switchport block unicast
```

You can verify your setting by entering the **show interfaces** *interface-id* **switchport** privileged EXEC command.

# system mtu

To set the global maximum packet size or MTU size for switched packets on Gigabit Ethernet and 10-Gigabit Ethernet ports, use the **system mtu** command in global configuration mode. To restore the global MTU value to its default value, use the **no** form of this command.

**system mtu** *bytes*
**no system mtu**

## Syntax Description

| | |
|---|---|
| *bytes* | The global MTU size in bytes. The range is 1500 to 9198 bytes; the default is 1500 bytes. |

## Command Default

The default MTU size for all ports is 1500 bytes.

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

You can verify your setting by entering the **show system mtu** privileged EXEC command.

The switch does not support the MTU on a per-interface basis.

If you enter a value that is outside the allowed range for the specific type of interface, the value is not accepted.

## Examples

This example shows how to set the global system MTU size to 6000 bytes:

```
Device(config)# system mtu 6000
Global Ethernet MTU is set to 6000 bytes.
Note: this is the Ethernet payload size, not the total
Ethernet frame size, which includes the Ethernet
header/trailer and possibly other tags, such as ISL or
802.1q tags.
```

# transport (coap-proxy configuration)

To configure transport protocol, use the **transport** command in coap-proxy configuration mode.

**transport**{**tcp** | **udp**}

**Syntax Description**

| | |
|---|---|
| **tcp** | Specifies a TCP protocol. |
| **udp** | Specifies a UDP protocol. |

**Command Modes**

coap-proxy configuration (config-coap-proxy)

**Command History**

| **Release** | **Modification** |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

**Usage Guidelines**

To access coap-proxy configuration mode, enter the **coap proxy** command in global configuration mode.

### Example

This is an example to configure tcp as transport protocol

```
Device(config)# coap proxy
Device(config-coap-proxy)# transport tcp
```

# voice-signaling vlan (network-policy configuration)

To create a network-policy profile for the voice-signaling application type, use the **voice-signaling vlan** command in network-policy configuration mode. To delete the policy, use the **no** form of this command.

**voice-signaling vlan** {*vlan-id* [{**cos** *cos-value* | **dscp** *dscp-value*}] | **dot1p** [{**cos** *l2-priority* | **dscp** *dscp*}] | **none** | **untagged**}

### Syntax Description

| | |
|---|---|
| *vlan-id* | (Optional) The VLAN for voice traffic. The range is 1 to 4094. |
| **cos** *cos-value* | (Optional) Specifies the Layer 2 priority class of service (CoS) for the configured VLAN. The range is 0 to 7; the default is 5. |
| **dscp** *dscp-value* | (Optional) Specifies the differentiated services code point (DSCP) value for the configured VLAN. The range is 0 to 63; the default is 46. |
| **dot1p** | (Optional) Configures the phone to use IEEE 802.1p priority tagging and to use VLAN 0 (the native VLAN). |
| **none** | (Optional) Does not instruct the Cisco IP phone about the voice VLAN. The phone uses the configuration from the phone key pad. |
| **untagged** | (Optional) Configures the phone to send untagged voice traffic. This is the default for the phone. |

### Command Default

No network-policy profiles for the voice-signaling application type are defined.

The default CoS value is 5.

The default DSCP value is 46.

The default tagging mode is untagged.

### Command Modes

Network-policy profile configuration

### Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

### Usage Guidelines

Use the **network-policy profile** global configuration command to create a profile and to enter network-policy profile configuration mode.

The voice-signaling application type is for network topologies that require a different policy for voice signaling than for voice media. This application type should not be advertised if all of the same network policies apply as those advertised in the voice policy TLV.

When you are in network-policy profile configuration mode, you can create the profile for voice-signaling by specifying the values for VLAN, class of service (CoS), differentiated services code point (DSCP), and tagging mode.

These profile attributes are contained in the Link Layer Discovery Protocol for Media Endpoint Devices (LLDP-MED) network-policy time-length-value (TLV).

To return to privileged EXEC mode from the network-policy profile configuration mode, enter the **exit** command.

This example shows how to configure voice-signaling for VLAN 200 with a priority 2 CoS:

```
(config)# network-policy profile 1
(config-network-policy)# voice-signaling vlan 200 cos 2
```

This example shows how to configure voice-signaling for VLAN 400 with a DSCP value of 45:

```
(config)# network-policy profile 1
(config-network-policy)# voice-signaling vlan 400 dscp 45
```

This example shows how to configure voice-signaling for the native VLAN with priority tagging:

```
(config-network-policy)# voice-signaling vlan dot1p cos 4
```

# voice vlan (network-policy configuration)

To create a network-policy profile for the voice application type, use the **voice vlan** command in network-policy configuration mode. To delete the policy, use the **no** form of this command.

**voice vlan** {*vlan-id* [{**cos** *cos-value* | **dscp** *dscp-value*}] | **dot1p** [{**cos** *l2-priority* | **dscp** *dscp*}] | **none** | **untagged**}

## Syntax Description

| | |
|---|---|
| *vlan-id* | (Optional) The VLAN for voice traffic. The range is 1 to 4094. |
| **cos** *cos-value* | (Optional) Specifies the Layer 2 priority class of service (CoS) for the configured VLAN. The range is 0 to 7; the default is 5. |
| **dscp** *dscp-value* | (Optional) Specifies the differentiated services code point (DSCP) value for the configured VLAN. The range is 0 to 63; the default is 46. |
| **dot1p** | (Optional) Configures the phone to use IEEE 802.1p priority tagging and to use VLAN 0 (the native VLAN). |
| **none** | (Optional) Does not instruct the Cisco IP phone about the voice VLAN. The phone uses the configuration from the phone key pad. |
| **untagged** | (Optional) Configures the phone to send untagged voice traffic. This is the default for the phone. |

## Command Default

No network-policy profiles for the voice application type are defined.

The default CoS value is 5.

The default DSCP value is 46.

The default tagging mode is untagged.

## Command Modes

Network-policy profile configuration

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Fuji 16.9.2 | This command was introduced. |

## Usage Guidelines

Use the **network-policy profile** global configuration command to create a profile and to enter network-policy profile configuration mode.

The voice application type is for dedicated IP telephones and similar devices that support interactive voice services. These devices are typically deployed on a separate VLAN for ease of deployment and enhanced security through isolation from data applications.

When you are in network-policy profile configuration mode, you can create the profile for voice by specifying the values for VLAN, class of service (CoS), differentiated services code point (DSCP), and tagging mode.

These profile attributes are contained in the Link Layer Discovery Protocol for Media Endpoint Devices (LLDP-MED) network-policy time-length-value (TLV).

To return to privileged EXEC mode from the network-policy profile configuration mode, enter the **exit** command.

This example shows how to configure the voice application type for VLAN 100 with a priority 4 CoS:

```
(config)# network-policy profile 1
(config-network-policy)# voice vlan 100 cos 4
```

This example shows how to configure the voice application type for VLAN 100 with a DSCP value of 34:

```
(config)# network-policy profile 1
(config-network-policy)# voice vlan 100 dscp 34
```

This example shows how to configure the voice application type for the native VLAN with priority tagging:

```
(config-network-policy)# voice vlan dot1p cos 4
```