Configuring Auto SmartPorts

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information about Auto SmartPorts

Auto SmartPort macros dynamically configure ports based on the device type detected on the port. When the switch detects a new device on a port, it applies the appropriate Auto SmartPorts macro. When a link-down event occurs on the port, the switch removes the macro. For example, when you connect a Cisco IP phone to a port, Auto SmartPorts automatically applies the Cisco IP phone macro. The Cisco IP phone macro enables quality of service (QoS), security features, and a dedicated voice VLAN to ensure proper treatment of delay-sensitive voice traffic.

Auto SmartPorts uses event triggers to map devices to macros. The most common event triggers are based on Cisco Discovery Protocol (CDP) messages received from connected devices. The detection of a device (Cisco IP phone, Cisco wireless access point, or Cisco router) invokes an event trigger for that device.

Note Although Auto SmartPort detects the Cisco switch it doesnot invoke the event trigger automatically. The event trigger needs to be manually invoked to map the switch to macros.

Link Layer Discovery Protocol (LLDP) is used to detect devices that do not support CDP. Other mechanisms used as event triggers include the 802.1X authentication result and MAC-address learned.

System built-in event triggers exist for various devices based mostly on CDP and LLDP messages and some MAC address. These triggers are enabled as long as Auto SmartPort is enabled.

You can configure user-defined trigger groups for profiles and devices. The name of the trigger group is used to associate a user-defined macro.

Auto SmartPort Macros

The Auto SmartPort macros are groups of CLI commands. Detection of devices on a port triggers the application of the macro for the device. System built-in macros exist for various devices, and, by default, system built-in triggers are mapped to the corresponding built-in macros. You can change the mapping of built-in triggers or macros as needed.

A macro basically applies or removes a set of CLIs on an interface based on the link status. In a macro, the link status is checked. If the link is up, then a set of CLIs is applied; if the link is down, the set is removed (the no format of the CLIs are applied). The part of the macro that applies the set of CLIs is termed macro. The part that removes the CLIs (the no format of the CLIs) are termed antimacro.

When a device is connected to an Auto SmartPort, if it gets classified as a lighting end point, it invokes the event trigger **CISCO_LIGHT_EVENT**, and the macro **CISCO_LIGHT_AUTO_SMARTPORT** is executed.

Related Topics

Commands executed by CISCO_LIGHT_AUTO_SMARTPORT

When the macro is executed, it runs a series of commands on the switch.

The commands that are executed by running the macro **CISCO_LIGHT_AUTO_SMARTPORT** are:

- switchport mode access
- switchport port-security violation restrict
- switchport port-security mac-address sticky
- switchport port-security
- power inline port poe-ha
- storm-control broadcast level 50.00
- storm-control multicast level 50.00
- storm-control unicast level 50.00

- spanning-tree portfast

- spanning-tree bpduguard enable

Enabling Auto SmartPort

Note   Auto SmartPort is disabled by default.

To disable Auto SmartPorts macros on a specific port, use the **no macro auto global processing** interface command before enabling Auto SmartPort globally.

To enable Auto SmartPort globally, use the **macro auto global processing** global configuration command.

To enable Auto SmartPorts, perform this task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **device classifier**
4. **macro auto global processing**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**device classifier** **Example:** Device(config)# **device classifier**	Enables the device classifier. Use **no device classifier** command to disable the device classifier.
Step 4	**macro auto global processing**	Enables Auto SmartPorts on the switch globally.

	Command or Action	Purpose
	Example: Device(config)# **macro auto global processing**	Use **no macro auto global processing** command to disable Auto SmartPort globally.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Related Topics

Configuring Mapping Between Event Triggers and Built-in Macros

Note   You need to perform this task when a Cisco switch is connected to the Auto SmartPort.

To map an event trigger to a built-in macros, perform this task:

Before you begin

You need to enable auto smartport macros globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **macro auto execute** *event trigger* **builtin** *built-in macro name*
4. **macro auto trigger** *event trigger*

5. **device** *device_ID*
6. **end**
7. **show shell triggers**
8. **show running-config**
9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** Switch> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Switch# **configure terminal**	Enters global configuration mode.
Step 3	**macro auto execute** *event trigger* **builtin** *built-in macro name* **Example:** Switch(config)# **macro auto execute CISCO_SWITCH_EVENT builtin CISCO_SWITCH_AUTO_SMARTPORT**	Specifies a user-defined event trigger and a macro name. This action configures mapping from an event trigger to a built-in Auto Smartports macro.
Step 4	**macro auto trigger** *event trigger* **Example:** Switch(config)# **macro auto trigger CISCO_SWITCH_EVENT**	Invokes the user-defined event trigger.
Step 5	**device** *device_ID* **Example:** Switch(config)# **device cisco WS-C3560CX-8PT-S**	Matches the event trigger to the device identifier.
Step 6	**end** **Example:** Switch(config)# **end**	Returns to privileged EXEC mode.
Step 7	**show shell triggers** **Example:** Switch# **show shell triggers**	Displays the event triggers on the switch.
Step 8	**show running-config** **Example:** Switch# **show running-config**	Verifies your entries.

	Command or Action	Purpose
Step 9	**copy running-config startup-config** **Example:** Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Example: Enabling Auto SmartPorts

This example shows how you can enable to Auto SmartPort.

```
Device> enable
Device# configure terminal
Device(config)# device classifier
Device(config)# macro auto global processing
Device(config)# end
```

Related Topics

Example: Configuring Mapping Between Event Triggers and Built-in Macros

This example shows how you can configure mapping between event triggers and built-in macros.

```
Switch> enable
Switch# configure terminal
Switch(config)# macro auto execute CISCO_SWITCH_EVENT builtin CISCO_SWITCH_AUTO_SMARTPORT
Switch(config)# macro auto trigger CISCO_SWITCH_EVENT
Switch(config)# device cisco WS-C3560CX-8PT-S
Switch(config)# end
```