



# CHAPTER 1

## Configuring Cisco IOS Auto Smartport Macros

---

This chapter describes how to configure and apply Auto Smartport macros on the Catalyst 4500 series switch.

This chapter includes the following major sections:

- [About Auto Smartport Macros, page 1-1](#)
- [Configuring Auto Smartport Macros, page 1-3](#)
- [Displaying Auto Smartport, page 1-14](#)



**Note**

For complete syntax and usage information for the switch commands used in this chapter, see the *Cisco Catalyst 4500 Series Switch Command Reference* and related publications at this location:

<http://www.cisco.com/en/US/products/hw/switches/ps4324/index.html>

If a command is not in the *Catalyst 4500 Series Switch Command Reference*, you can locate it in the Cisco IOS library. See the *Cisco IOS Command Reference* and related publications at this location:

<http://www.cisco.com/en/US/products/ps6350/index.html>

---

## About Auto Smartport Macros

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 Smartport 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 Smartport 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 Smartport 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 does not 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 (Table 1-1) and some MAC address. (Through Cisco IOS Release 12.2(54)SG, DMP is detected using the MAC address. Starting with Cisco IOS Release 15.0(2)SG, DMP is also detected using CDP.) These triggers are enabled as long as Auto Smartport is enabled.

You can also define your own trigger. User-defined triggers can be CDP/LLDP-based, a group of MAC addresses, or the value of the attribute-value (AV) pair for the **auto-smart-port** keyword.

The Auto Smartport macros are groups of CLI commands. Detection of devices on a port triggers the application of the macro for the device. (For example, detecting a CISCO\_PHONE event on a port triggers the switch to apply the commands in the CISCO\_PHONE\_AUTO\_SMARTPORT macro.) 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*.

Besides creating user-defined triggers, you can also create user-defined macros and map one to the other among all triggers (both built-in and user-defined) and all macros (both built-in and user-defined). Use the Cisco IOS scripting capability to create the macros. Cisco IOS scripting is a BASH-like language syntax for command automation and variable replacement.

The four detection mechanisms adhere to the following order of priority:

- If 802.1X authentication is configured on a port, an authentication response-based trigger is applied, and other triggers are ignored.
- If 802.1X authentication fails and the CDP/LLDP fallback mechanism is configured, CDP/LLDP triggers for phone devices only; if no fallback mechanism is configured, or a device is not a phone device, nothing is triggered.
- If 802.1X authentication is configured on a port, a MAC address-based trigger is never triggered.
- If 802.1X authentication is not configured on a port, CDP/LLDP has priority over a MAC address-based trigger with a hold-off timer applied for MAC-address based trigger. Between CDP/LLDP, there is no particular order; whichever one arrives first is triggered.

## Device Classifier

Starting with Cisco Release IOS XE 3.3.0SG and IOS 15.1(1)SG, the device classifier (DC) feature is enabled by default on the Catalyst 4500 series switch.

The DC collects information from MAC-OUI and protocols such as CDP, LLDP, and DHCP to identify devices. You must enable CDP and LLDP on the switch. To make DHCP options information available to the DC, you must enable the DHCP snooping feature on the switch. The device attributes collected from these protocols are evaluated against a set of profiles available to the DC to find the best match. The best-matched profile is used for device identification.

Devices that do not send CDP, LLDP or DHCP traffic may not be properly identified by the device classifier.

Device-classifier uses profile definitions—built-in and default profiles. The built-in profiles contain the device profiles that are known to the Auto Smartport module, comprising a limited set of Cisco devices. They are built into Cisco IOS and cannot be changed. The default profiles are stored as a text file in nonvolatile storage and allow the DC to identify a much larger set of devices. The default profiles are updated as part of the Cisco IOS archive download.

When a new device is detected, the corresponding shell trigger executes the Auto Smartport configuration macro. Auto Smartport has built-in mappings for a large set of devices. You can use the commands described in the [“Configuring Mapping Between User-Defined Triggers and Built-in Macros” section on page 1-10](#) to create new mappings. You can create the trigger mappings based on the profile name or device name that is provided by the DC.

## Device Visibility Mode

The DC function is enabled on the switch by default. You can disable it by using the **no macro auto monitor** global configuration command. The DC feature provides **show** commands to display the devices that are connected to the switch. It also provides information about the physical port to which the device is connected, along with device MAC address and other vendor information. Only directly connected devices, such as another Layer 2 switch, are classified on nonaccess ports. On access ports that are connected to hubs, device classification is limited to 32 devices.

When you enable Auto Smartport, the DC is automatically enabled.

# Configuring Auto Smartport Macros

The following topics are included:

- [Enabling Auto Smartport Macros, page 1-3](#)
- [Auto Smartport Configuration Guidelines, page 1-5](#)
- [Configuring Auto Smartport Built-in Macro Parameters, page 1-6](#)
- [Configuring Mapping Between Event Triggers and Built-in Macros, page 1-8](#)
- [Configuring User-Defined Event Triggers, page 1-9](#)
- [Configuring Mapping Between User-Defined Triggers and Built-in Macros, page 1-10](#)
- [Configuring Auto Smartport User-Defined Macros, page 1-11](#)

## Enabling Auto Smartport Macros



### Note

---

By default, Auto Smartport is disabled globally. To disable Auto Smartport 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 Smartport macros, perform this task:

	Command	Purpose
Step 1	Switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	Switch(config)# <b>[no] macro auto global processing [cdp   lldp]</b>	Enables Auto Smartport on the switch globally.  <b>Note</b> Starting with Release 15.0(2)SG, the <b>fallback</b> option has been deprecated.  Use <b>no macro auto global processing</b> to disable Auto Smartport globally.  <b>Note</b> The <b>macro auto processing</b> command turns Auto Smartport on or off on the interface level. The default is on.
Step 3	Switch(config)# <b>end</b>	Returns to privileged EXEC mode.
Step 4	Switch# <b>show running-config</b>	Verifies that Auto Smartport is enabled.
Step 5	Switch# <b>copy running-config startup-config</b>	(Optional) Saves your entries in the configuration file.

Use the **show shell functions** and the **show shell triggers** privileged EXEC command to display the event triggers, the built-in macros, and the built-in macro default values.

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

```
Switch(config)# macro auto global processing
Switch(config)# interface interface_id
Switch(config-if)# no macro auto processing
```

## Auto Smartport Default Configuration

By default, Cisco IOS shell is enabled and Auto Smartport is disabled globally.

Table 1-1 shows the Auto Smartport built-in event triggers that are embedded in the switch software by default.

**Table 1-1 Auto Smartport Built-in Event Trigger Macros**

Event Trigger Name	Description
CISCO_PHONE_EVENT	System detects that a phone device is connected to an interface.
CISCO_SWITCH_EVENT	System detects that a switch is connected to an interface.
CISCO_ROUTER_EVENT	System detects that a router is connected to an interface.
CISCO_WIRELESS_AP_EVENT	System detects that a wireless application is connected to an interface.
CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT	System detects that a wireless lightweight application is connected to an interface.
CISCO_DMP_EVENT	System detects that a digital media player is connected to an interface.
CISCO_IPVSC_EVENT	System detects that an IP video surveillance camera is connected to an interface.

Table 1-2 shows the Auto Smartport built-in macros that are embedded in the switch software.

**Table 1-2 Auto Smartport Built-in Macros**

Macro Name	Description
CISCO_PHONE_AUTO_SMARTPORT	Use this macro for Cisco IP phone device. It enables QoS, port security, Address Resolution Protocol (ARP) inspection (dynamic ARP inspection), IP source guard, DHCP snooping, storm control and spanning tree protection on the port.
CISCO_SWITCH_AUTO_SMARTPORT	Use this macro to apply the switch macro for Cisco switches. It enables trunking on the port.
CISCO_ROUTER_AUTO_SMARTPORT	Use this macro to apply the router macro for Cisco routers. It enables QoS, trunking, and spanning-tree protection on the port.
CISCO_AP_AUTO_SMARTPORT	Use this macro to apply the wireless access point (AP) macro for Cisco APs. It enables support for an autonomous wireless access point and QoS on the port.
CISCO_LWAP_AUTO_SMARTPORT	Use this macro to apply the lightweight wireless access point macro for Cisco lightweight wireless APs. It enables QoS, port security, dynamic ARP inspection, IP source guard, DHCP snooping, storm control, and spanning tree protection on the port.
CISCO_IP_CAMERA_AUTO_SMARTPORT	Use this macro for a Cisco IP surveillance camera device. It enables QoS, port security, and access VLAN on the port.
CISCO_DMP_AUTOSMARTPORT	Use this macro for a Cisco digital media player device. It enables QoS, port security, and access VLAN on the port.



**Note**

By default, the built-in event triggers are mapped to the built-in macros.

## Auto Smartport Configuration Guidelines

Auto Smartport guidelines include the following:

- To avoid system conflicts when Auto Smartport macros are applied, remove all port configuration except for 802.1X authentication.
- If the macro conflicts with the original configuration, some macro commands might not be applied, or some antimacro commands might not be applied. (The antimacro is the portion of the applied macro that removes it at link down.)



**Note**

Failure of one command in the macro halts the application of the entire macro.

For example, if 802.1X authentication is enabled, you cannot remove switchport-mode access configuration. You must remove the 802.1X authentication before removing the configuration.

- A port should not be a member of an EtherChannel when applying Auto Smartport macros. If Auto Smartport is not yet enabled globally, disable Auto Smartport on all the EtherChannel ports before enabling it globally. If Auto Smartport is already enabled, shut down the port and disable it before adding the port to an EtherChannel.



**Note** If an Auto Smartport macro is applied on an interface, EtherChannel configuration usually fails because of conflict with the auto-QoS configuration applied by the macro.

- The built-in macro default data VLAN is VLAN 1. The default voice VLAN is VLAN 2. You should modify the built-in macro default values if your switch uses different VLANs. To view all built-in macro default values, use the **show shell functions** privileged EXEC command.
- To detect non-Cisco devices for 802.1X authentication or MAB, configure the RADIUS server to support the Cisco AV pair **auto-smart-port=event trigger**. You must configure a user-defined trigger with the value returned in the AV pair for **auto-smart-port**.
- For stationary devices that do not support CDP, MAB, or 802.1X authentication, such as network printers, we recommend that you disable Auto Smartport on the port.
- If authentication is enabled on a port, the switch ignores CDP unless the **fallback cdp** keyword is in the **macro auto global processing** global configuration command.
- The order of CLI commands within the macro and the corresponding antimacro can differ.
- Before converting a port into an Layer 3 interface, enter the **no macro auto processing** command. This prevents Auto Smartport from applying macros on the interface. If Layer 3 is already configured, enter the **no macro auto processing** command on the Layer 3 interface enable Auto Smartport globally.
- Auto Smartport macros and Smartport cannot coexist on an interface.
- A switch applies a macro in accordance with the LLDP advertisement from the attached device. If the device does not identify itself properly, the wrong macro is applied. Consult the specific device documentation to ensure the device's firmware is current.
- The LWAP's WLC software version must be 6.0.188 ( => Cisco IOS 12.4(21a)JA2) or later to make it detectable as LWAP by AutoSmartport.
- As of Cisco IOS Release 12.2(54)SG, Auto Smartport does not support macros that apply EtherChannel configurations. Interfaces that belong to EtherChannel groups are treated as standard interfaces. You can apply macros on individual interfaces based on the device type but the CLIs in the macro (for example, auto-QoS) might conflict with an EtherChannel configuration. We recommend that you disable Auto Smartport on interfaces belonging to EtherChannels before you enable Auto Smartport globally. If Auto Smartport is already enabled, disable Auto Smartport on the interfaces before configuring EtherChannel.
- When a Cisco switch is detected on the Auto Smartport, you have to manually map the event trigger to either a built-in macro or user-defined macro. You need to also match the event trigger to the device PID.

## Configuring Auto Smartport Built-in Macro Parameters

The switch automatically maps from built-in event triggers to built-in macros. You can replace the built-in macro default values with values that are specific to your switch.

To configure Auto Smartport built-in macros parameters, perform this task:

	Command	Purpose
Step 1	Switch# <b>configure terminal</b>	Enters global configuration mode.

	Command	Purpose
<b>Step 2</b>	<pre>Switch(config)# <b>macro auto execute</b> <b>event trigger builtin</b> <i>built-in</i> <b>macro name</b> [<i>parameter=value</i>] [<i>parameter=value</i>]</pre>	<p>Defines mapping from an event trigger to a built-in macro.</p> <p>Specify an <i>event trigger</i> value:</p> <ul style="list-style-type: none"> <li>• CISCO_PHONE_EVENT</li> <li>• CISCO_SWITCH_EVENT</li> <li>• CISCO_ROUTER_EVENT</li> <li>• CISCO_WIRELESS_AP_EVENT</li> <li>• CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT</li> <li>• CISCO_DMP_EVENT</li> <li>• CISCO_IPVSC_EVENT</li> <li>• WORD—Apply a user-defined event trigger.</li> </ul> <p>Specify a <i>built-in macro name</i> value:</p> <ul style="list-style-type: none"> <li>• CISCO_PHONE_AUTO_SMARTPORT (Optional) Specify the parameter values: \$ACCESS_VLAN=(1) and \$VOICE_VLAN=(2).</li> <li>• CISCO_SWITCH_AUTO_SMARTPORT (Optional) Specify the parameter values: \$NATIVE_VLAN=(1).</li> <li>• CISCO_ROUTER_AUTO_SMARTPORT (Optional) Specify the parameter values: \$NATIVE_VLAN=(1).</li> <li>• CISCO_AP_AUTO_SMARTPORT (Optional) Specify the parameter values: \$NATIVE_VLAN=(1).</li> <li>• CISCO_LWAP_AUTO_SMARTPORT (Optional) Specify the parameter values: \$ACCESS_VLAN=(1).</li> <li>• CISCO_DMP_AUTO_SMARTPORT</li> <li>• CISCO_IP_CAMERA_AUTO_SMARTPORT</li> </ul> <p>(Optional) <i>parameter=value</i>—Replace default values that begin with \$. Enter new values in the form of name value pair separated by a space: [<i>name1=value1 name2=value2...</i>]. Default values are shown in parenthesis.</p>
<b>Step 3</b>	<pre>Switch(config)# <b>end</b></pre>	Returns to privileged EXEC mode.
<b>Step 4</b>	<pre>Switch# <b>show running-config</b></pre>	Verifies your entries.
<b>Step 5</b>	<pre>Switch# <b>copy running-config</b> <b>startup-config</b></pre>	(Optional) Saves your entries in the configuration file.

The **no macro auto execute event trigger** {[**builtin** *built-in macro name* [*parameter=value*]] | [[*parameter=value*] {*function contents*}} command deletes the mapping.

This example shows how to use two built-in Auto Smartport 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:

```
Switch# configure terminal
Switch(config)# macro auto execute CISCO_PHONE_EVENT builtin CISCO_PHONE_AUTO_SMARTPORT
ACCESS_VLAN=10 VOICE_VLAN=20
Switch(config)#
Switch(config)#
```

```

Switch(config)#!!! the next command enables auto smart ports globally
Switch(config)# macro auto global processing fallback cdp
Switch(config)#
Switch(config)# exit

Switch#
Switch# show running-config interface gigabitethernet2/7
Building configuration...

Current configuration : 284 bytes
!
switchport access vlan 10
switchport mode access
switchport voice vlan 2
switchport port-security maximum 2
switchport port-security
switchport port-security aging time 2
switchport port-security violation restrict
switchport port-security aging type inactivity
auto qos voip cisco-phone
qos trust device cisco-phone
neighbor device type phone
macro description CISCO_PHONE_EVENT
spanning-tree portfast
spanning-tree bpduguard enable
service-policy input AutoQos-VoIP-Input-Cos-Policy
service-policy output AutoQos-VoIP-Output-Policy
end

```

**Note**

You can also use the **macro auto device** command to simplify changing the parameters for a built-in functions for a device type.

## 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 event trigger to a built-in macros, perform this task:

	Command	Purpose
Step 1	Switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	Switch(config)# <b>macro auto execute event trigger builtin built-in macro name</b>	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 3	Switch(config)# <b>macro auto trigger event trigger</b>	Invokes the user-defined event trigger.
Step 4	Switch(config)# <b>device device_ID</b>	Matches the event trigger to the device identifier.
Step 5	Switch(config)# <b>end</b>	Returns to privileged EXEC mode.
Step 6	Switch# <b>show shell triggers</b>	Displays the event triggers on the switch.
Step 7	Switch# <b>copy running-config startup-config</b>	(Optional) Saves your entries in the configuration file.



This example shows how to map an event trigger called CISCO\_SWITCH\_EVENT to the built-in macro CISCO\_SWITCH\_AUTO\_SMARTPORT.

```
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)# exit
```

## Configuring User-Defined Event Triggers

You can configure two types of event triggers: user-defined and MAC address-based.

The following sections describe these triggers:

- [802.1X-Based Event Trigger, page 1-9](#)
- [MAC Address-Based Event Trigger, page 1-10](#)

### 802.1X-Based Event Trigger

When using MAB or 802.1X authentication to trigger Auto Smartport macros, you need to create an event trigger that corresponds to the Cisco AV pair (**auto-smart-port=event trigger**) sent by the RADIUS server.

To configure an event trigger, perform this task:

	Command	Purpose
Step 1	Switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	Switch(config)# <b>shell trigger</b> <i>identifier description</i>	Specifies the event trigger identifier and description. The identifier should have no spaces or hyphens between words.
Step 3	Switch(config)# <b>end</b>	Returns to privileged EXEC mode.
Step 4	Switch# <b>show shell triggers</b>	Displays the event triggers on the switch.
Step 5	Switch# <b>copy running-config</b> <b>startup-config</b>	(Optional) Saves your entries in the configuration file.

Use the **no shell trigger identifier** global configuration command to delete the event trigger.

The following example shows how to define a user-defined trigger:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# shell trigger RADIUS_MAB_EVENT MAC_AuthBypass Event
Switch(config)#
```

## MAC Address-Based Event Trigger

To configure a MAC address group as an event trigger, perform this task:

	Command	Purpose
Step 1	Switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	Switch(config)# <b>macro auto mac-address group</b>	Specifies a group of MAC address as an event trigger. Changes mode to config-mac-addr-grp. You can then add or remove the MAC address or Organizational Unique Identifier (OUI) from the group. The <i>group</i> value defines the user-defined trigger.
Step 3	Switch(config)# <b>end</b>	Returns to privileged EXEC mode.
Step 4	Switch# <b>show shell triggers</b>	Displays the event triggers on the switch.
Step 5	Switch# <b>copy running-config startup-config</b>	(Optional) Saves your entries in the configuration file.

Use the **no macro auto mac-address-group grp\_name** to delete the event trigger.

## Configuring Mapping Between User-Defined Triggers and Built-in Macros

You need to map the user-defined trigger to either a built-in macro or user-defined macro.

To map a user-defined trigger to a built-in macros, perform this task:

	Command	Purpose
Step 1	Switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	Switch(config)# <b>macro auto execute event trigger builtin built-in macro name [parameter=value] [parameter=value]</b>	Specifies a user-defined event trigger and a macro name. This action replaces built-in macro default values, and configures mapping from an event trigger to a built-in Auto Smartport macros. <b>Note</b> When performing a mapping, you must provide parameter values. For example, you must specify \$ACCESS_VLAN=(1) and \$VOICE_VLAN=(2) for the macro CISCO_PHONE_AUTO_SMARTPORT.
Step 3	Switch(config)# <b>end</b>	Returns to privileged EXEC mode.
Step 4	Switch# <b>show shell triggers</b>	Displays the event triggers on the switch.
Step 5	Switch# <b>copy running-config startup-config</b>	(Optional) Saves your entries in the configuration file.

This example shows how to map a user-defined event trigger called RADIUS\_MAB\_EVENT to the built-in macro CISCO\_PHONE\_AUTO\_SMARTPORT with access VLAN set to 10, and how to verify the entries.

This procedure shows how to map a user-defined trigger to a built-in macro:

- 
- Step 1** Connect the device to a MAB-enabled switch port.
  - Step 2** On the RADIUS server, set the attribute-value pair to auto-smart-port=RADIUS\_MAB\_EVENT.

**Step 3** On the switch, create the event trigger RADIUS\_MAB\_EVENT.

The switch recognizes the attribute-value pair=RADIUS\_MAB\_EVENT response from the RADIUS server and applies the macro CISCO\_PHONE\_AUTO\_SMARTPORT, as in the following example:

```
Switch(config)# macro auto execute RADIUS_MAB_EVENT builtin CISCO_PHONE_AUTO_SMARTPORT
ACCESS_VLAN=10
Switch(config)# exit
Switch# show shell triggers
User defined triggers
-----
Trigger Id: RADIUS_MAB_EVENT
Trigger description: MAC_AuthBypass Event
Trigger environment:
Trigger mapping function: CISCO_PHONE_AUTO_SMARTPORT
<output truncated>
```

## Configuring Auto Smartport User-Defined Macros

The Cisco IOS shell provides basic scripting capabilities for configuring the user-defined Auto Smartport macros. These macros can contain multiple lines and can include any CLI command. You can also define variable substitution, conditionals, functions, and triggers within the macro.

Inside a user-defined macro, besides parameters specified through **macro auto execute trigger parameter-name=value ..**, you also can use the following variables published by EEM ([Table 1-3](#)):

**Table 1-3** Variables Published by EEM

Parameter Name	Meaning
\$INTERFACE	Name of the interface where the trigger event is detected.
\$LINKUP	Indicates whether the interface is up or down (true/false).
\$TRIGGER	Name of the trigger event that is raised (for example, CISCO_PHONE_EVENT).
\$AUTH_ENABLED	Indicates whether 802.1X authentication is configured on the interface (true/false).

To map an event trigger to a user-defined macro, perform this task:

	Command	Purpose
Step 1	Switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	Switch(config)# <b>macro auto execute event trigger</b> [parameter=value] {function contents}	Specifies a user-defined macro that maps to an event trigger.  Specify an <i>event trigger</i> value: <ul style="list-style-type: none"> <li>• CISCO_PHONE_EVENT</li> <li>• CISCO_SWITCH_EVENT</li> <li>• CISCO_ROUTER_EVENT</li> <li>• CISCO_WIRELESS_AP_EVENT</li> <li>• CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT</li> <li>• WORD Applies a user-defined event trigger.</li> <li>• CISCO_DMP_EVENT</li> <li>• CISCO_IPVSC_EVENT</li> </ul> <i>function contents</i> —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.  (Optional) <i>parameter=value</i> —Replaces default values that begin with \$, enter new values in the form of name value pair separated by a space: [name1=value1 name2=value2...].
Step 3	Switch(config)# <b>end</b>	Returns to privileged EXEC mode.
Step 4	Switch# <b>show running-config</b>	Verifies your entries.
Step 5	Switch# <b>copy running-config startup-config</b>	(Optional) Saves your entries in the configuration file.

This example shows how to map a user-defined event trigger called Cisco Digital Media Player (DMP) to a user-defined macro.

- 
- Step 1** Connect the DMP to an 802.1X- or MAB-enabled switch port.
  - Step 2** On the RADIUS server, set the attribute-value pair to **auto-smart-port=MY\_MEDIAPLAYER\_EVENT**.
  - Step 3** On the switch, create the event trigger CISCO\_DMP\_EVENT, and map it to the user-defined macro commands shown below.

The switch recognizes the attribute-value pair=CISCO\_DMP\_EVENT response from the RADIUS server and applies the macro associated with this event trigger.

The following example shows the macro portion of the automacro:

```
Switch(config)# shell trigger CISCO_DMP_EVENT Cisco DMP player
Switch(config)# macro auto execute CISCO_DMP_EVENT {
if [[ $LINKUP -eq 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

```

The following represents the anti-macro portion of the automacro:

```

if [[ $LINKUP -eq NO ]]; then
conf t
interface $INTERFACE
    no macro description $TRIGGER
    no switchport access vlan 1
    if [[ $AUTH_ENABLED -eq 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
}
Switch(config)# end

```

Table 1-4 lists the supported shell keywords you can apply in your macros and antimacro statements.

**Table 1-4 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.
-eq	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 1-5 lists the shell keywords that are not supported in macros and antimacros.

**Table 1-5** *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.
time	Pipeline.
until	Looping construct.
while	Looping construct.

## Displaying Auto Smartport

To display the Auto Smartport and static Smartport macros, use one or more of the privileged EXEC commands in Table 1-6.

**Table 1-6** *Commands for Displaying Auto Smartport and Static Smartport Macros*

Command	Purpose
<code>show macro auto monitor clients</code>	Displays the clients using the device classifier facility on the switch.
<code>show macro auto monitor device</code>	Displays the devices connected to a switch and their associated properties.
<code>show macro auto monitor type</code>	Displays all the device types recognized by the device classifier.
<code>show parser macro</code>	Displays all static Smartport macros.
<code>show parser macro name <i>macro-name</i></code>	Displays a specific static Smartport macro.
<code>show parser macro brief</code>	Displays the static Smartport macro names.
<code>show parser macro description [interface <i>interface-id</i>]</code>	Displays the static Smartport macro description for all interfaces or for a specified interface.
<code>show shell</code>	Displays information about Auto Smartport event triggers and macros.

This example shows how to use the `show macro auto monitor device` privileged EXEC command with the optional `mac-address` keyword to view summary information about the connected device with the specified MAC address:

```
Switch# show macro auto monitor device 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 macro auto monitor type` privileged EXEC command with no optional keywords to view the devices recognized by the device classifier:

```

Switch# show macro auto monitor 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

This example shows how to use the **show shell triggers** privileged EXEC command to view the event triggers in the switch software:

```
Switch# show shell triggers
```

```
User defined triggers
```

```
-----
```

```
Built-in triggers
```

```
-----
```

```
Trigger Id: CISCO_PHONE_EVENT
```

```
Trigger description: Event for ip-phone macro
```

```
Trigger environment: ACCESS_VLAN=1 VOICE_VLAN=2
```

```
Trigger mapping function: CISCO_PHONE_AUTO_SMARTPORT
```

```
Trigger Id: CISCO_ROUTER_EVENT
```

```
Trigger description: Event for router macro
```

```
Trigger environment: NATIVE_VLAN=1
```

```
Trigger mapping function: CISCO_ROUTER_AUTO_SMARTPORT
```

```
Trigger Id: CISCO_SWITCH_EVENT
```

```
Trigger description: Event for switch macro
```

```
Trigger environment: NATIVE_VLAN=1
```

```
Trigger mapping function: CISCO_SWITCH_AUTO_SMARTPORT
```

```
Trigger Id: CISCO_WIRELESS_AP_EVENT
```

```
Trigger description: Event for Wireless Access Point macro
```

```
Trigger environment: NATIVE_VLAN=1
```

```
Trigger mapping function: CISCO_AP_AUTO_SMARTPORT
```

```
Trigger Id: CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT
```

```
Trigger description: Event for Wireless Lightweight Access Point macro
```

```
Trigger environment: NATIVE_VLAN=1
```



Trigger mapping function: CISCO\_LWAP\_AUTO\_SMARTPORT

This example shows how to use the **show shell functions** privileged EXEC command to view the built-in macros in the switch software:

```
Switch# show shell functions
#User defined functions:

#Built-in functions:
function CISCO_AP_AUTO_SMARTPORT () {
    if [[ $LINKUP -eq 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
            exit
        end
    fi
    if [[ $LINKUP -eq 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
                if [[ $AUTH_ENABLED -eq NO ]]; then
                    no switchport mode
                    no switchport trunk encapsulation
                fi
            fi
        exit
    end
fi
}

function CISCO_SWITCH_AUTO_SMARTPORT () {
    if [[ $LINKUP -eq YES ]]; then
        conf t
            interface $INTERFACE
                macro description $TRIGGER
                auto qos voip trust
                switchport trunk encapsulation dot1q
                switchport trunk native vlan $NATIVE_VLAN
                switchport trunk allowed vlan ALL
                switchport mode trunk
            exit
        end
    else
        conf t
            interface $INTERFACE
                no macro description
                no auto qos voip trust
                no switchport mode trunk
                no switchport trunk encapsulation dot1q
                no switchport trunk native vlan $NATIVE_VLAN
                no switchport trunk allowed vlan ALL
            exit
        end
    fi
}
```

## ■ Displaying Auto Smartport

```
        end
    fi
}
<output truncated>
```