



CHAPTER 2

Configuring Auto Smartports and Static Smartports Macros

- [Configuring Macros, page 2-1](#)
- [Displaying Macros, page 2-27](#)

Configuring Macros

- [Auto Smartports Configuration Guidelines, page 2-1](#)
- [Enabling Auto Smartports Macros, page 2-2](#)
- [Default Auto Smartports Configuration, page 2-3](#)
- [Configuring Auto Smartports Parameter Values, page 2-6](#)
- [Configuring MAC Address Groups, page 2-7](#)
- [Configuring Macro Persistence, page 2-8](#)
- [Configuring Built-In Macro Options, page 2-9](#)
- [Creating User-Defined Event Triggers, page 2-12](#)
- [Configuring Event Trigger Control, page 2-16](#)
- [Configuring User-Defined Macros, page 2-18](#)
- [Applying Macros on a Switch, page 2-22](#)
- [Default Static Smartports Configuration, page 2-25](#)
- [Static Smartports Configuration Guidelines, page 2-25](#)
- [Applying Static Smartports Macros, page 2-26](#)

Auto Smartports Configuration Guidelines

- You cannot 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** global configuration 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 you enable device-specific Auto Smartports on the switch. The switch applies the appropriate port-based commands.
- 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.

- The built-in-macro default data VLAN is VLAN 1. The built-in macro default voice VLAN is VLAN 2. (VLAN 1 is the default data VLAN for all macros. VLAN 2 is the default voice VLAN for all macros.) If your switch uses different access, native, or voice VLANs, use the **macro auto device** or the **macro auto execute** global configuration commands to configure the values.
- If you use the VLAN name in a macro, it must be the same name that is in the VLAN database for all switches in the VLAN Trunking Protocol (VTP) domain.
- Use the **show macro auto device** privileged EXEC command to display the default macros with the default parameter values, current values, and the configurable parameter list for each macro. You can also use the **show shell functions** privileged EXEC command to see the built-in-macro default values.
- To use 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**.
- For stationary devices that do not support CDP, MAB, or 802.1x authentication, such as network printers, you can configure a MAC address group with a MAC OUI-based trigger and map it to a user-defined macro with the desired configuration.
- An 802.1x-authentication-based trigger takes precedence over all other event triggers, such as Cisco Discovery Protocol (CDP) messages, Link Layer Discovery Protocol (LLDP) messages, or user-defined MAC address groups.
- The switch supports Auto Smartport macros only on directly connected devices. Multiple device connections, such as hubs, are not supported. If multiple devices are connected, the applied macro is associated with the first detected device.
- If authentication is enabled on a port, the switch ignores a MAC address trigger if authentication fails, and there is no fallback for MAC address trigger support.
- The order of CLI commands within the macro and the corresponding antimacro can be different.
- Lightweight access point in Remote Edge Access Point (REAP) or Hybrid Remote Edge Access Point (HREAP) modes are not supported by Auto Smartports.

Enabling Auto Smartports Macros

Follow this required procedure to enable macros globally on the switch.

Beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	macro auto global processing	Globally enables macros on the switch.
Step 3	end	Returns to privileged EXEC mode.
Step 4	show running-config	Verifies that Auto Smartports is enabled.
Step 5	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return to the default setting, use the **no macro auto global processing** global configuration command.

To disable macros on a specific port, use the **no macro auto processing** interface configuration command.

You can use the **show macro auto device**, the **show shell functions**, and the **show shell triggers** privileged EXEC commands to display the event triggers and the built-in macros.

This example shows how to enable macros on the switch and then how to disable macros on a specific interface:

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

Default Auto Smartports Configuration

- Auto Smartports is globally disabled and is enabled per interface.
- CDP fallback is globally disabled and is enabled per interface.
- Macro persistence is globally disabled and per interface.
- Cisco IOS shell is enabled.
- The switch uses these built-in macros (the defaults) when Auto Smartports is enabled for the specific devices.

Table 2-1 *Device-Specific Built-In Macros*

Macro Name	Description
CISCO_AP_AUTO_SMARTPORT	This macro applies the wireless access point macro for Cisco access points. It enables standard QoS, auto-QoS, and 802.1q encapsulated trunking. It also configures the native VLAN on the interface.
CISCO_DMP_AUTO_SMARTPORT	This macro applies the digital media player macro for Cisco digital media players. It enables QoS trust, auto-QoS, port security, and spanning-tree protection. It configures the access VLAN for the interface and provides network protection from unknown unicast packets.
CISCO_IPVSC_AUTO_SMARTPORT	This macro applies the IP camera macro for Cisco IP video surveillance cameras. It enables QoS trust, auto-QoS, port security, and spanning-tree protection. It configures the access VLAN for the interface and provides network protection from unknown unicast packets.

Table 2-1 *Device-Specific Built-In Macros (continued)*

Macro Name	Description
CISCO_LWAP_AUTO_SMARTPORT	This macro applies the lightweight wireless access point macro for Cisco lightweight wireless access points. It enables QoS, port security, storm control, DHCP snooping, and spanning-tree protection. It configures the access VLAN for the interface and provides network protection from unknown unicast packets.
CISCO_PHONE_AUTO_SMARTPORT	This macro applies the IP phone macro for Cisco IP phones. It enables QoS, port security, storm control, DHCP snooping, and spanning-tree protection. It also configures the access and voice VLANs for that interface.
CISCO_ROUTER_AUTO_SMARTPORT	This macro applies the router macro for Cisco routers. It enables QoS and trunking with 802.1Q encapsulation and spanning-tree bridge protocol data unit (BPDU) protection.
CISCO_SWITCH_AUTO_SMARTPORT	This macro applies the switch macro for Cisco switches. It enables QoS and trunking with 802.1q encapsulation. It also configures the native VLAN on the interface.

Use these macros when the switch does not use device-specific macros.

Table 2-2 *Global and Custom Macros*

Macro Name	Description
CISCO_CUSTOM_AUTO_SMARTPORT	This macro applies the per-port user-defined settings after the antimacro ¹ is applied on a switch port. You specify the settings in the macro.
CISCO_LAST_RESORT_AUTO_SMARTPORT	This macro applies a per-port device-specific macro when the switch does not have built-in macro for the device. It has a basic configuration with a data VLAN.
CISCO_SWITCH_AAA_ACCOUNTING	This macro applies the authentication, authorization, and accounting (AAA) accounting settings.
CISCO_SWITCH_AAA_AUTHENTICATION	This macro applies the authentication, authorization, and accounting (AAA) authentication settings.
CISCO_SWITCH_AAA_AUTHORIZATION	This macro applies the authentication, authorization, and accounting (AAA) authorization settings.
CISCO_SWITCH_AUTO_IP_CONFIG	This macro applies the IP settings
CISCO_SWITCH_AUTO_PCI_CONFIG	This macro applies Payment Card Industry (PCI)-compliant settings.
CISCO_SWITCH_DOMAIN_NAME_CONFIG	This macro applies the domain name.
CISCO_SWITCH_ETHERCHANNEL_CONFIG	This macro applies the EtherChannel settings.
CISCO_SWITCH_HOSTNAME_CONFIG	This macro applies the hostname.
CISCO_SWITCH_HTTP_SERVER_CONFIG	This macro applies the HTTP server settings.
CISCO_SWITCH_LOGGING_SERVER_CONFIG	This macro applies the logging server settings.
CISCO_SWITCH_MGMT_VLAN_CONFIG	This macro applies the management VLAN settings.

Table 2-2 Global and Custom Macros (continued)

Macro Name	Description
CISCO_SWITCH_NAME_SERVER_CONFIG	This macro applies the name server settings.
CISCO_SWITCH_NTP_SERVER_CONFIG	This macro applies the Network Time Protocol (NTP) server settings.
CISCO_SWITCH_RADIUS_SERVER_CONFIG	This macro applies the RADIUS server settings.
CISCO_SWITCH_SETUP_SNMP_TRAPS	This macro applies the Simple Network Management Protocol (SNMP) trap settings.
CISCO_SWITCH_SETUP_USR_CONFIG	This macro applies user settings.
CISCO_SWITCH_SNMP_SOURCE_CONFIG	This macro applies the SNMP source interface settings.
CISCO_SWITCH_TACACS_SERVER_CONFIG	This macro applies the TACACS server settings.
CISCO_SWITCH_USER_PASS_CONFIG	This macro applies the username and password settings.

1. The antimacro is the portion of the applied macro that removes the macro at a link-down event.

In Cisco IOS Release 12.2(55)SE

- The switch applies the CISCO_PHONE_AUTO_SMARTPORT macro to these IP phones:
 - Cisco IP Phone 8961
 - Cisco IP Phone 9951
 - Cisco IP Phone 9971
- The access point macros have these enhancements:
 - The switch uses the platform string in the CDP message to determine the access point type (autonomous or lightweight) and then applies the appropriate macro.
 - To reduce overrun errors at the ingress interface on an access point Ethernet receiver, the switch adds the QoS bandwidth setting to the access point macros when it receives a CDP message with the auto-QoS type, length, and value attributes (TLVs). QoS derives the bandwidth value from the auto-QoS TLVs.

If the CDP messages does not have the auto-QoS TLVs, the switch does not add the bandwidth setting to the macros.



Note

If you do not upgrade the access point image to one that has the auto-QoS TLVs, the switch does not add the bandwidth setting to the access point macros. When you configure the bandwidth before the link to the receiver goes down, the setting is removed when the link comes up.

If you add a macro command that sets the QoS bandwidth and the switch applies the macro to an access point that does not support the auto-QoS TLVs, the command is not applied to the access point. We recommend that you create a user-defined macro without that command.

- When a Catalyst 3750-E and 3560-E switch is connected to a Cisco Aironet 1250 access point, the switch applies a power setting to allocate up to 20 watts (W).

Configuring Auto Smartports Parameter Values

The switch automatically maps from event triggers to built-in device-specific macros. You can follow this optional procedure to replace macro default parameter values with values that are specific to your switch.

Beginning in privileged EXEC mode:

	Command	Purpose
Step 1	show macro auto device	Displays the macro default parameter values.
Step 2	configure terminal	Enters global configuration mode.
Step 3	macro auto device {access-point ip-camera lightweight-ap media-player phone router switch} [parameter=value]	Replaces the specified macro default parameter values. Enter new values in the form of a name-value pair separated by spaces: [<name1>=<value1> <name2>=<value2>...]. You can enter the VLAN ID or the VLAN name when specifying VLAN parameter values. Default values are shown for each macro default parameter value. • access-point NATIVE_VLAN=1 • ip-camera ACCESS_VLAN=1 • lightweight-ap ACCESS_VLAN=1 • media-player ACCESS_VLAN=1 • phone ACCESS_VLAN=1 VOICE_VLAN=2 • router NATIVE_VLAN=1 • switch NATIVE_VLAN=1 Note You must enter the correct parameter name (for example, VOICE_VLAN) because this text string must match the text string in the built-in macro definition.
Step 4	end	Returns to privileged EXEC mode.
Step 5	show macro auto device	Verifies your entries.
Step 6	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return to the default setting, use the **no macro auto device {macro name} parameter=value** global configuration command.

This example shows how to see the IP phone macro parameter values and how to change the default voice VLAN to 20. When you change the default values, they are not immediately applied on the interfaces with existing applied macros. The configured values are applied at the next link-up event. Note that the exact text string was used for VOICE_VLAN. The entry is case sensitive.

```
Switch# show macro auto device phone
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

Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# macro auto device phone VOICE_VLAN=20
Switch(config)# end
Switch# show macro auto device phone
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:voice_vlan=20

```

Configuring MAC Address Groups

For devices such as printers that do not support neighbor discovery protocols such as CDP or LLDP, use the MAC-address-based trigger configurations. This optional procedure requires these steps:

1. Configure a MAC-address-based trigger by using the **macro auto mac-address** global configuration command.
2. Associate the MAC address trigger to a built-in or a user-defined macro by using the **macro auto execute** global configuration command.

Beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	macro auto mac-address-group <i>name</i>	Specifies the group name, and enter MAC address configuration mode.
Step 3	[mac-address list <i>list</i>] [oui [<i>list</i> <i>list</i> <i>range</i> <i>start-value</i> <i>size</i> <i>number</i>]]	Configures a list of MAC addresses separated by spaces. Specify an operationally unique identifier (OUI) list or range . The OUI is the first three bytes of the MAC address and identifies the manufacturer of the product. Specifying the OUI allows devices that do not support neighbor discovery protocols to be recognized. • list—Enter an OUI list in hexadecimal format separated by spaces. • range—Enter the starting OUI hexadecimal value (<i>start-value</i>). • size—Enter the length of the range (<i>number</i>) from 1 to 5 to create a list of sequential addresses.
Step 4	exit	Returns to configuration mode.
Step 5	macro auto execute <i>address_trigger</i> built-in <i>macro name</i>	Maps the MAC address-group trigger to a built-in or user-defined macro. The MAC address trigger is applied to an interface after 65 seconds. The switch uses this hold time to apply a CDP- or LLDP-based event trigger instead of the MAC address trigger.
Step 6	end	Returns to privileged EXEC mode.
Step 7	show macro auto address-group	Verifies your entries.
Step 8	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To delete an address group, use the **no macro auto mac-address-group** *name* global configuration command. Enter **no macro auto mac-address-group** *name* to remove the macro trigger and any associated trigger that maps to a macro defined by the **macro auto execute** global configuration command. Entering **no macro auto execute mac-address-group** only removes the mapping of the trigger to the macro.

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

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# macro auto mac-address-group address_trigger
Switch(config-addr-grp-mac)# mac-address list 2222.3333.3334 22.33.44 a.b.c
Switch(config-addr-grp-mac)# oui list 455555 233244
Switch(config-addr-grp-mac)# oui range 333333 size 2
Switch(config-addr-grp-mac)# exit
Switch(config)# macro auto execute address_trigger builtin CISCO_PHONE_AUTO_SMARTPORT
Switch(config)# end
Switch# show running configuration | include macro
macro auto mac-address-group address_trigger
mac auto mac-address-group hel
mac auto execute address_trigger builtin CISCO_PHONE_AUTO_SMARTPORT
  macro description CISCO_DMP_EVENT
  mac description CISCO_SWITCH_EVENT
!
```

<output truncated>

The example shows how to create an OUI list with five sequential addresses starting with 00000A and how to verify your entries:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# macro auto mac-address-group size5ouilist
Switch(config-addr-grp-mac)# oui range 00000A size 5
Switch(config-addr-grp-mac)# exit
Switch(config)# mac auto execute size5-ouilist builtin macro
Switch(config)# macro auto execute address_trigger builtin CISCO_PHONE_AUTO_SMARTPORT
Switch(config)# end
Switch# show running configuration | include oui
oui list 00000E
oui list 00000D
oui list 00000C
oui list 00000B
oui list 00000A
```

Configuring Macro Persistence

When you enable Auto Smartports on the switch, by default the macro configuration is applied at a link-up event and removed at a link-down event. When you enable macro persistence, the configuration is applied at link-up and is not removed at link-down. The applied configuration remains. Macro persistence remains configured after a reboot if you have saved the running configuration file.

Follow this optional procedure so that enable macros remain active on the switch after a link-down event.

Beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	interface <i>interface-id</i>	Specifies an interface and enters interface configuration mode.
Step 3	macro auto sticky	Enables macros to remain active on the interface after a link-down event.
Step 4	end	Returns to privileged EXEC mode.
Step 5	show running-config interface <i>interface-id</i>	Verifies your entries.
Step 6	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable macro persistence, use the **no macro auto sticky** interface configuration command.

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

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet 2/0/1
Switch(config-if)# macro auto port sticky
Switch(config-if)# exit
Switch(config)# end
Switch# show running-config interface gigabitethernet 2/0/1
Building configuration...

Current configuration : 243 bytes
!
<output truncated>
!
interface GigabitEthernet2/0/1
 srr-queue bandwidth share 1 30 35 5
 queue-set 2
 priority-queue out
 mls qos trust device cisco-phone
 mls qos trust cos
 macro auto port sticky
  service-policy input AUTOQOS-ENHANCED-CISCOPHONE-POLICY
end

<output truncated>

Switch#
```

Configuring Built-In Macro Options

Use this procedure to map event triggers to built-in macros and to replace the built-in macro default parameters with values that are specific to your switch. If you need to *replace* default parameters values in a macro, use the **macro auto device** global configuration command. All commands in this procedure are optional.

Beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	macro auto execute <i>event trigger</i> builtin <i>built-in macro name</i> [<i>parameter=value</i>] [<i>parameter=value</i>]	Defines mapping from an event trigger to a built-in macro. Specify an <i>event trigger</i>: • CISCO_CUSTOM_EVENT • CISCO_DMP_EVENT • CISCO_IPVSC_EVENT • CISCO_LAST_RESORT_EVENT • CISCO_PHONE_EVENT • CISCO_ROUTER_EVENT • CISCO_SWITCH_EVENT • CISCO_WIRELESS_AP_EVENT • CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT • WORD—Apply a user-defined event trigger. Specify a builtin <i>built-in macro name</i>: Enter new values in the form of <i>name value pair</i> separated by spaces: [<i><name1>=<value1> <name2>=<value2>...</i>]. Default values are shown exactly as they should be entered. • CISCO_AP_AUTO_SMARTPORT Specify the parameter value: NATIVE_VLAN=1. • CISCO_DMP_AUTO_SMARTPORT Specify the parameter value: ACCESS_VLAN=1. • CISCO_IPVSC_AUTO_SMARTPORT Specify the parameter value: ACCESS_VLAN=1. • CISCO_LWAP_AUTO_SMARTPORT Specify the parameter value: ACCESS_VLAN=1. • CISCO_PHONE_AUTO_SMARTPORT Specify the parameter values: ACCESS_VLAN=1 and VOICE_VLAN=2. • CISCO_ROUTER_AUTO_SMARTPORT Specify the parameter value: NATIVE_VLAN=1. • CISCO_SWITCH_AUTO_SMARTPORT Specify the parameter value: NATIVE_VLAN=1.

	Command	Purpose
Step 3	remote url	Specifies a remote server location for the remote macro file: - The syntax for the local flash file system on the standalone switch or the stack master: flash: - The syntax for the local flash file system on a stack member: flash member number: - The syntax for the FTP: ftp:[[/username[:password]@location]/directory]/filename - The syntax for an HTTP server: http:[[/username:password]@]{hostname host-ip}[/directory]/filename - The syntax for a secure HTTP server: https:[[/username:password]@]{hostname host-ip}[/directory]/filename - The syntax for NVRAM: nvr:[[/username:password]@]/[directory]/filename - The syntax for the Remote Copy Protocol (RCP): rcp:[[/username@location]/directory]/filename - The syntax for the Secure Copy Protocol (SCP): scp:[[/username@location]/directory]/filename - The syntax for the TFTP: tftp:[[/location]/directory]/filename
Step 4	end	Returns to privileged EXEC mode.
Step 5	show running-config	Verifies your entries.
Step 6	copy running-config startup-config	Saves your entries in the configuration file.

This example shows how to use two built-in macros to connect 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
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#!!! the next command modifies the access and voice vlans
Switch(config)#!!! for the built in Cisco IP phone auto smartport macro
Switch(config)# macro auto execute CISCO_PHONE_EVENT builtin CISCO_PHONE_AUTO_SMARTPORT
ACCESS_VLAN=10 VOICE_VLAN=20
Switch(config)#
Switch(config)#!!! the next command modifies the Native vlan used for inter switch trunks
Switch(config)# macro auto execute CISCO_SWITCH_EVENT builtin CISCO_SWITCH_AUTO_SMARTPORT
NATIVE_VLAN=10
Switch(config)#
Switch(config)#!!! the next command enables auto smart ports globally
Switch(config)# macro auto global processing cdp-fallback
Switch(config)#
Switch(config)# exit

Switch# !!! here is the running configuration of the interface connected
Switch# !!! to another Cisco Switch after the Macro is applied
Switch#
Switch# show running-config interface gigabitethernet1/0/1
Building configuration...
```

```

Current configuration : 284 bytes
!
interface GigabitEthernet1/0/1
interface GigabitEthernet0/1
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 configure the remote macro for native VLAN 5.

- a. Configure the remote macro in the macro.txt file.
- b. Use the **macro auto execute** configuration command to specify the remote location for the macro file.

Macro.txt file

```

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
end

```

```

Switch(config)# macro auto execute CISCO_SWITCH_EVENT remote tftp://<ip_address>/macro.txt
NATIVE_VLAN=5

```

```

Switch# show running configuration | include macro
macro auto execute CISCO_SWITCH_EVENT remote tftp://<ip_address>/macro.txt
NATIVE_VLAN=5
Switch#

```

Creating User-Defined Event Triggers

When using MAB or 802.1x authentication as an event trigger, create a trigger that corresponds to the Cisco attribute-value pair (**auto-smart-port=event trigger**) sent by the RADIUS server. This procedure is optional.

Beginning in privileged EXEC mode:

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

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

This example shows how to map a user-defined event trigger called RADIUS_MAB_EVENT to the built-in macro CISCO_AP_AUTO_SMARTPORT, to replace the default VLAN with VLAN 10, and how to verify the entries.

- Connect the device to a MAB-enabled switch port.
- On the RADIUS server, set the attribute-value pair to **auto-smart-port=RADIUS_MAB_EVENT**.
- 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_AP_AUTO_SMARTPORT.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# !!! create a user defined trigger and map
Switch(config)# !!! a system defined macro to it
Switch(config)# !!! first create the trigger event
Switch(config)# shell trigger RADIUS_MAB_EVENT MAC_AuthBypass Event
Switch(config)#
Switch(config)#!!! map a system defined macro to the trigger event
Switch(config)# macro auto execute RADIUS_MAB_EVENT builtin ?
CISCO_AP_AUTO_SMARTPORT      Configure native vlan and trust cos
CISCO_CUSTOM_AUTOSMARTPORT   Configure user defined parameters
CISCO_DMP_AUTO_SMARTPORT     Configure access vlan, qos and port-security
CISCO_IP_CAMERA_AUTO_SMARTPORT Configure access vlan, qos and port-security
CISCO_LAST_RESORT_SMARTPORT   Configure access vlan
CISCO_LWAP_AUTO_SMARTPORT     Configure native vlan, qos, port-security and
                                storm-control
CISCO_PHONE_AUTO_SMARTPORT    Configure access vlan, voice vlan, trust
                                device, interface bandwidth, port-security
CISCO_ROUTER_AUTO_SMARTPORT   Configure native vlan, spanning tree
                                port-fast,trunk mode and trust dscp
CISCO_SWITCH_AUTO_SMARTPORT   Configure native vlan, trunk mode
Switch(config)# macro auto execute RADIUS_MAB_EVENT builtin CISCO_AP_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_AP_SMARTPORT
<output truncated>
```

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_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=(),
                    $SEC_MODE=(), $SEC_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** 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
                if [[ $LIMIT -eq 0 ]]; then
                    default srr-queue bandwidth limit
                else
                    srr-queue bandwidth limit $LIMIT
                fi
                if [[ $SW_POE -eq YES ]]; then
                    if [[ $AP125X -eq AP125X ]]; then
                        macro description AP125X
                        macro auto port sticky
                        power inline port maximum 20000
                    fi
                fi
            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
                default srr-queue bandwidth limit
                if [[ $AUTH_ENABLED -eq NO ]]; then
                    no switchport mode
                    no switchport trunk encapsulation
                fi
                if [[ $STICKY -eq YES ]]; then
                    if [[ $SW_POE -eq YES ]]; then
                        if [[ $AP125X -eq AP125X ]]; then
                            no macro auto port sticky
                            no power inline port maximum
                        fi
                    fi
                fi
            fi
        fi
    fi
}
```

```

        fi
    exit
end
fi
}
<output truncated>

```

Configuring Event Trigger Control

Use event trigger control to specify when the switch applies macros. By default, the switch maps built-in and user-defined macros to these triggers:

- Detection method (for example, MAC address groups, MAB messages, 802.1x authentication messages, and LLDP messages)
- Device type (for example, Cisco switches, routers, and IP phones)
- Configured triggers

When you select triggers, the switch applies macros only when those triggers map to macros.

On a Switch

Beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	macro auto global control { detection [cdp] [lldp] [mac-address] device [access-point] [ip-camera] [lightweight-ap] [media-player] [phone] [router] [switch] trigger [last-resort] }	Specifies when the switch applies a macro based on the detection method, device type, or trigger. • detection—Use one or more of these as an event trigger: – cdp—CDP messages – lldp—LLDP messages – mac-address—User-defined MAC address groups • device—Use one or more of these devices as an event trigger: – access-point—Autonomous access point – ip-camera—Cisco IP video surveillance camera – lightweight-ap—Lightweight access point – media-player—Digital media player – phone—Cisco IP phone – router—Cisco router – switch—Cisco switch • trigger—Use a specific event trigger. – (Optional) last-resort—Last-resort trigger. By default, the switch uses all detection method, device types, and configured triggers.

	Command	Purpose
Step 3	end	Returns to privileged EXEC mode.
Step 4	show running-config	Verifies your entries.
Step 5	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no macro auto global control** global configuration command to disable a specific trigger-to-macro mapping. The switch does not apply the macros mapped to the specific triggers.

To configure the switch to apply the CISCO_AP_AUTO_SMARTPORT macro only when it detects an autonomous access point:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# macro auto global control device access-point
Switch(config)# end
```

To configure the switch to apply the CISCO_AP_AUTO_SMARTPORT or the CISCO_SWITCH_AUTO_IP_CONFIG macro only when it detects an autonomous access point or a Cisco IP phone:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# macro auto global control device access-point phone
Switch(config)# end
```

On an Interface

Beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	interface <i>interface_id</i>	Specifies an interface and enters interface configuration mode.

	Command	Purpose
Step 3	macro auto control { detection [cdp] [lldp] [mac-address] device [access-point] [ip-camera] [lightweight-ap] [media-player] [phone] [router] [switch] trigger [last-resort] }	Specifies when the switch applies a macro based on the detection method, device type, or trigger. • detection—Use one or more of these as an event trigger: – cdp—CDP messages – lldp—LLDP messages – mac-address—User-defined MAC address groups • device—Use one or more of these devices as an event trigger: – access-point—Autonomous access point – ip-camera—Cisco IP video surveillance camera – lightweight-ap—Lightweight access point – media-player—Digital media player – phone—Cisco IP phone – router—Cisco router – switch—Cisco switch • trigger—Use a specific event trigger. – (Optional) last-resort—Last-resort trigger. By default, the switch uses all detection method, device types, and configured triggers.
Step 4	exit	Returns to global configuration mode.
Step 5	end	Returns to privileged EXEC mode.
Step 6	show macro auto	Verifies your entries.
Step 7	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no macro auto global control** global configuration command to disable a specific trigger-to-macro mapping. The switch does not apply the macros mapped to the specific triggers.

To configure the switch to apply the CISCO_AP_AUTO_SMARTPORT or CISCO_PHONE_AUTO_SMARTPORT macro only when it detects a lightweight access point or a Cisco IP phone:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet 5/0/1
Switch(config-if)# macro auto control device lightweight-ap phone
Switch(config-if)# exit
Switch(config)# end
```

Configuring User-Defined Macros

The Cisco IOS shell has basic scripting capabilities for configuring user-defined 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. This procedure is optional.

**Note**

When configuring macros, you must enter a description. If the link is down (command **\$LINKUP -eq NO**), you must enter the **no macro description** command. These commands are mandatory for Auto Smartports to work.

Beginning in privileged EXEC mode, follow these steps to map a user-defined event trigger to a user-defined macro.

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	macro auto execute <i>event trigger</i> [<i>parameter=value</i>] { <i>function contents</i> }	Specifies a user-defined macro that maps to an event trigger. { <i>function contents</i> } Specify 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> —Replace default values that begin with \$, and enter new values in the form of name value pair separated by spaces: [<i><name1>=<value1> <name2>=<value2>...</i>].
Step 3	end	Returns to privileged EXEC mode.
Step 4	show running-config	Verifies your entries.
Step 5	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Example: User-Defined Event Trigger and Macro

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

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

```
Switch(config)# shell trigger MP_EVENT mediaplayer
Switch(config)# macro auto execute MP_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
```

```

if [[ $LINKUP -eq NO ]]; then
conf t
interface $INTERFACE
    no macro description
    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

```

Example: Last-Resort Event Trigger and Macro

To map the CISCO_LAST_RESORT_AUTO_SMARTPORT macro to the last-resort trigger:

```
Switch(config)# macro auto global control trigger last-resort
```

CISCO_LAST_RESORT_AUTO_SMARTPORT macro:

```

if [[ $LINKUP -eq YES ]]; then
    conf t
        interface $INTERFACE
            macro description $TRIGGER
            switchport access vlan $ACCESS_VLAN
            switchport mode access
            spanning-tree portfast
            spanning-tree bpdufilter enable
            load-interval 60
            no shutdown
        exit
    end
fi
if [[ $LINKUP -eq NO ]]; then
    conf t
        interface $INTERFACE
            no macro description
            no switchport access vlan $ACCESS_VLAN
            no switchport mode access
            no spanning-tree portfast
            no spanning-tree bpdufilter enable
            no load-interval 60
        exit
    end
fi

```

Example of user-defined mapping of a MAC address trigger to map a last-resort macro:

```

Switch(config)#macro auto mac
Switch(config)#macro auto mac-address-group Laptop
Switch(config-addr-grp-mac)#mac-address list 0000.0011.2233
Switch(config-addr-grp-mac)#exit
Switch(config)#macro auto execute laptop builtin CISCO_LAST_RESORT_SMARTPORT
ACCESS_VLAN=10

```

```
Switch(config)#end
Switch#
```

Example: Custom Event Trigger and CISCO_CUSTOM_AUTO_SMARTPORT Macro

Default CISCO_CUSTOM_AUTO_SMARTPORT macro:

```
if [[ $LINKUP -eq YES ]]; then
    conf t
        interface $INTERFACE
        exit
    end
fi
if [[ $LINKUP -eq NO ]]; then
    conf t
        interface $INTERFACE
        exit
    end
fi
```

To create a user-defined macro with the same name as the custom macro, override the CISCO_CUSTOM_AUTO_SMARTPORT macro, and set the parameters for your switch, including the mapping from an event trigger to the macro.

```
Config# macro auto execute CISCO_CUSTOM_EVENT {
    if [[ $LINKUP -eq YES ]]; then
        conf t
            interface $INTERFACE
            description asp3-link-UP i.e. Custom Macro OFF
            no macro description
            switchport
            switchport mode access
            switchport access vlan $ACCESS_VLAN
            spanning-tree portfast
            exit
        end
    fi
    if [[ $LINKUP -eq NO ]]; then
        conf t
            interface $INTERFACE
            macro description $TRIGGER
            switchport access vlan $ACCESS_VLAN
            description asp3-link-DOWN i.e. Custom Macro ON
            exit
        end
    fi
}
```

Table 2-3 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.

Table 2-3 *Supported Cisco IOS Shell Keywords (continued)*

Command	Description
-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 2-4 *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.

Applying Macros on a Switch

You can use the CLI or the Cisco IOS shell scripting capability to set the macro parameters and to apply the macro.

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

Using the CLI

Beginning in privileged EXEC mode:

	Command	Purpose
Step 1	macro auto config ?	(Optional) Displays the global macros.
Step 2	macro auto config <i>global macro</i>	Sets the macro parameters. Follow the prompts in the CLI.
Step 3	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Example: One Macro

To display the global macros:

```
Switch# 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

Switch# 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

Switch# macro auto config CISCO_SWITCH_HOSTNAME_CONFIG
Enter system's network name: CISCO
Do you want to apply the parameters? [yes/no]: yes
Enter configuration commands, one per line.  End with CNTL/Z.
Switch# macro auto apply CISCO_SWITCH_HOSTNAME_CONFIG
Enter configuration commands, one per line.  End with CNTL/Z.
CISCO#
```

Example: Combined Macros

```
Switch# macro auto config CISCO_SWITCH_AUTO_IP_CONFIG
Do you want to configure default domain name? [yes/no]: yes
Enter the domain name: cisco.com
Do you want to configure Name server ipv4 address? [yes/no]: yes
Enter the IPv4 address[a.b.c.d]: 10.77.11.34
Enter IP address of the logging host: 10.77.11.36
```

```

Do you want to configure VPN Routing/Forwarding Instance name? [yes/no]: no
Enter the ip address of NTP server[a.b.c.d]: 10.77.11.37
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.
Switch# macro auto apply CISCO_SWITCH_AUTO_IP_CONFIG
Enter configuration commands, one per line. End with CNTL/Z.
Switch#

```

Using the Cisco IOS Shell

Beginning in privileged EXEC mode:

	Command	Purpose
Step 1	macro auto config ? or macro auto apply ?	(Optional) Displays the global macros.
Step 2	macro auto config <i>macro-name</i> <i>parameter=value</i> [<i>parameter=value</i>] ...	Sets the macro parameters. Follow the prompts in the CLI.
Step 3	macro auto apply <i>macro-name</i>	Applies the macro to the switch.
Step 4	show macro auto	Verifies your entries. The user-defined values appear only in the show command output.
Step 5	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Example: One Single Shell Parameter for One Macro

```

Switch# macro auto config CISCO_SWITCH_HOSTNAME_CONFIG HOSTNAME=CISCO
Switch# macro auto apply CISCO_SWITCH_HOSTNAME_CONFIG
Enter configuration commands, one per line. End with CNTL/Z.
CISCO#

```

Example: Multiple Shell Parameters and Values for One Macro

```

Switch# macro auto config CISCO_SWITCH_ETHERCHANNEL_CONFIG PORT_CH_ID=1 PORT_CH_TYPE=2  
EC_PROTO=Y EC_PROTO_TYPE=PAGP NO_OF_INT=3 MODE=AUTO INTERFACE=Gig2/0/1,Gig2/0/2,Gig2/0/3  
NON_SILENT=Y EC_APPLY=YES
Switch# macro auto apply CISCO_SWITCH_ETHERCHANNEL_CONFIG
Enter configuration commands, one per line. End with CNTL/Z.
Switch#

```

Example: Combined Macros

```

Switch# macro auto config CISCO_SWITCH_AUTO_IP_CONFIG CISCO_SWITCH_DOMAIN_NAME_CONFIG  
DOMAIN_NAME=cisco.com
Switch# macro auto config CISCO_SWITCH_AUTO_IP_CONFIG CISCO_SWITCH_LOGGING_SERVER_CONFIG  
HOST_IP=10.77.11.36
Switch# macro auto config CISCO_SWITCH_AUTO_IP_CONFIG CISCO_SWITCH_NAME_SERVER_CONFIG  
IP_V4_ADDR=10.77.11.37

```



```
Switch# macro auto config CISCO_SWITCH_AUTO_IP_CONFIG CISCO_SWITCH_NTP_SERVER_CONFIG
IP_ADDRESS=10.77.11.38 VRF=NO
Switch# macro auto apply CISCO_SWITCH_AUTO_IP_CONFIG
Enter configuration commands, one per line. End with CNTL/Z.
Switch#
```

Default Static Smartports Configuration

There are no static Smartports macros enabled on the switch.

Table 2-5 *Default Static Smartports Macros*

Macro Name ¹	Description
cisco-global	Use this global configuration macro to enable rapid per-VLAN spanning-tree plus (PVST+), loop guard, and dynamic port-error recovery for link state failures.
cisco-desktop	Use this interface configuration macro for increased network security and reliability when connecting a desktop device, such as a PC, to a switch port.
cisco-phone	Use this interface configuration macro when connecting a desktop device such as a PC with a Cisco IP Phone to a switch port. This macro is an extension of the cisco-desktop macro and provides the same security and resiliency feature and also dedicated voice VLANs to ensure proper treatment of delay-sensitive voice traffic.
cisco-switch	Use this interface configuration macro when connecting an access switch and a distribution switch or between access switches connected through small form-factor pluggable (SFP) modules.
cisco-router	Use this interface configuration macro when connecting the switch and a WAN router.
cisco-wireless	Use this interface configuration macro when connecting the switch and a wireless access point.

1. Cisco-default Smartports macros vary, depending on the software version running on your switch.

Static Smartports Configuration Guidelines

- When a macro is applied globally to a switch or to a switch interface, the existing configuration on the interface is retained. This is helpful when applying an incremental configuration.
- If a command fails because of a syntax or a configuration error, the macro continues to apply the remaining commands. You can use the **macro global trace macro-name** global configuration command or the **macro trace macro-name** interface configuration command to apply and then debug the macro to find any syntax or configuration errors.
- Some CLI commands are specific to certain interface types. If you apply a macro to an interface that does not accept the configuration, the macro fails the syntax or the configuration check, and the switch returns an error message.
- Applying a macro to an interface range is the same as applying a macro 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.
- When you apply a macro to a switch or a switch interface, the macro name is automatically added to the switch or interface. You can display the macro names and applied commands using the **show running-config** user EXEC command.

Applying Static Smartports Macros

Beginning in privileged EXEC mode, follow these steps to apply a static Smartports macro:

	Command	Purpose
Step 1	show parser macro	Displays the Cisco-default static Smartports macros embedded in the switch software.
Step 2	show parser macro name <i>macro-name</i>	Displays the specific macro that you want to apply.
Step 3	configure terminal	Enters global configuration mode.
Step 4	macro global { apply trace } <i>macro-name</i> [parameter { <i>value</i> }] [parameter { <i>value</i> }] [parameter { <i>value</i> }]	Applies a macro on the switch: - To only apply each individual macro command, use the macro global apply <i>macro-name</i> command. - To apply and then debug a macro to find any syntax or configuration errors, use the macro global trace <i>macro-name</i>. Append the macro with the required values by using the parameter value keywords. Keywords that begin with \$ require a unique parameter value. You can use the macro global apply <i>macro-name</i> ? command to display a list of any required values for the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied. (Optional) Specify unique parameter values that are specific to the switch. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive. The corresponding value replaces all matching occurrences of the keyword.
Step 5	interface <i>interface-id</i>	(Optional) Specifies an interface and enters interface configuration mode.
Step 6	default interface <i>interface-id</i>	(Optional) Clears all configuration from the specified interface.
Step 7	macro { apply trace } <i>macro-name</i> [parameter { <i>value</i> }] [parameter { <i>value</i> }] [parameter { <i>value</i> }]	Applies a macro on the interface: - To only apply each individual macro command, use the macro apply <i>macro-name</i> command. - To apply and then debug a macro to find any syntax or configuration errors, use the macro trace <i>macro-name</i>. Append the macro with the required values by using the parameter value keywords. Keywords that begin with \$ require a unique parameter value. You can use the macro global apply <i>macro-name</i> ? command to display a list of any required values for the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied. (Optional) Specify unique parameter values that are specific to the switch. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive. The corresponding value replaces all matching occurrences of the keyword.

	Command	Purpose
Step 8	end	Return to privileged EXEC mode.
Step 9	show running-config interface <i>interface-id</i>	Verify that the macro is applied to the interface.
Step 10	copy running-config startup-config	(Optional) Save your entries in the configuration file.

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

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

```
Switch# sh parser macro name cisco-desktop
-----
Macro name : cisco-desktop
Macro type : default interface
# macro keywords $access_vlan
# Basic interface - Enable data VLAN only
# Recommended value for access vlan should not be 1
switchport access vlan $access_vlan
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
-----
Switch#
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet1/0/4
Switch(config-if)# macro apply cisco-desktop $access_vlan 25
```

Displaying Macros

Table 2-6 Commands for Displaying Auto Smartports and Static Smartports Macros

Command	Purpose
show macro auto ?	Displays information about Auto Smartports macros. • device: Displays device macro information • event: Displays macro event-related commands • global: Displays global macro information • interface: Displays interface Auto Smartports status
show parser macro	Displays all static smartports macros.

Table 2-6 **Commands for Displaying Auto Smartports and Static Smartports Macros (continued)**

Command	Purpose
show parser macro name <i>macro-name</i>	Displays a specific static Smartports macro.
show parser macro brief	Displays the static Smartports macro names.
show parser macro description [interface <i>interface-id</i>]	Displays the static Smartports macro description for all interfaces or for a specified interface.
show shell	Displays information about Auto Smartports event triggers and macros. • data-path: Displays data paths for <i>fetch</i> • environment: Displays shell environment information • functions: Displays shell functions information • triggers: Displays shell triggers information