Configuring Auto Smartports and Static Smartports Macros

Configuring Macros

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. Be sure to enter the commands that remove port security and Bridge Protocol Data Unit (BPDU) guard features *before* you enable Auto Smartports macros on a port.

- 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.

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

- The built-in macro default data VLAN is VLAN 1. The built-in macro default voice VLAN is VLAN 2. If your switch uses different access, native, or voice VLANs, use the **macro auto device** or the **macro auto execute** global configuration commands to configure the values.

- You can specify either the VLAN name or the VLAN Id in a macro. If you use the VLAN name in a macro, it must be the same name as that in the VLAN database for all switches in the VLAN Trunking Protocol (VTP) domain. If you use the VLAN Id in a macro, then you must enter the correct VLAN name, else the default VLAN is applied.

- 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, LLDP, MAB, or 802.1x authentication, such as network printers, 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 Smartports macros only on directly connected devices. If multiple devices are connected, (for example, through a hub) 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.

- When using MAC-address-based detection, ensure that Auto Smartports is enabled *only* on ports facing access devices and not on ports that face the network or an intermediate gateway switch.

- The order of CLI commands within the macro and the corresponding antimacro can be different.

- When the device identity is configured and the device is authenticated on a switch port, these RADIUS attributes could be downloaded:

  – VLAN ID and switch ACL name or number from the Cisco access control server (ACS)

  – ASP trigger name in an attribute-value (AV) pair.

    After the AV pair is downloaded, the switch applies the macro on the port.

  The downloaded VLAN ID or ACL name could conflict with the settings in the user-defined or builtin macro applied by the switch.

- Auto Smartports does not support lightweight access points in the Remote Edge Access Point (REAP) or Hybrid Remote Edge Access Point (HREAP) mode.

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

- 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.

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** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**macro auto global processing** **Example:** `Switch(config)# macro auto global processing`	Globally enables macros on the switch.
Step 3	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

	Command	Purpose
Step 4	**show running-config** **Example:** `Switch# show running-config`	Verifies that Auto Smartports is enabled.

Step 5	**copy running-config startup-config**
	
	Example:
	`Switch# 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.

Note   The **no macro auto global processing** does not remove the macro which is already applied to the switch.

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.

- Macro persistence is globally disabled and is enabled per interface.

- Cisco IOS shell is disabled. Execute the Terminal shell EXEC command to enable IOS shell.

- The switch uses these built-in macros (the defaults) when Auto Smartports is enabled for the specific devices.

Table 3-1   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 configures the native VLAN on the interface. It also enables macro persistence so that the macro remains active after a link-down event.
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. **Note** If you enter the **auto qos video media-player** interface configuration command, the switch automatically uses the CDP to detect the presence or absence of a Cisco digital media player.
CISCO_IP_CAMERA_AU TO_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.
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_S MARTPORT	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.

- The access point macros have these enhancements:

  – The switch determines 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 3850 switch is connected to a Cisco Aironet 1250 access point, the switch applies a power setting to allocate up to 20 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** **Example:** `Switch# show macro auto device`	Displays the macro default parameter values.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**macro auto device** {**access-point** | **ip-camera** | **lightweight-ap** | **media-player** | **phone** | **router** | **switch**} [*parameter=value*] **Example:** `Switch(config)# macro auto device router`	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** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

	Command	Purpose
Step 5	**show macro auto device** **Example:** `Switch# show macro auto device`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

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** **Example:** `Switch# configure terminal`	Enters global configuration mode.

Step 2	**macro auto mac-address-group** *name*
	
	Example:
	`Switch(config)# macro auto`
	`mac-address-group address_trigger`
		Specifies the group name, and enter MAC address configuration mode.

Step 3	**mac-address list** *list* | [**oui** [**list** *list* |
	range *start-value* **size** *number*]]
	
	Example:
	`Switch(config-addr-grp-mac)#`
	`mac-address list 2222.3333.3334`
	`22.33.44 a.b.c`
		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 (*start-value*).
		
		• **size**—Enter the length of the range (*number*) from 1 to 5 to create a list of sequential addresses.

Step 4	**exit** **Example:** `Switch(mac-adress-config) exit`	Returns to configuration mode.

Step 5	**macro auto execute** *address_trigger*
	built-in *macro name*
	
	Example:
	`Switch(config)# macro auto execute`
	`address_trigger builtin`
	`CISCO_PHONE_AUTO_SMARTPORT`
		Maps the MAC address-group trigger to a built-in or user-defined macro.

Step 6	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

	Command	Purpose
Step 7	**show macro auto address-group** *name* **Example:** `Switch# show macro auto address-group group2`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to create a MAC-address-group event trigger called *address_trigger*, map it to the built in phone macro, and 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 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 size5ouilist builtin macro
Switch(config)# macro auto execute size5ouilist 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** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Switch(config)# interface gigabitethernet 2/0/1`	Specifies an interface and enters interface configuration mode.
Step 3	**macro auto sticky** **Example:** `Switch(config-if)# macro auto port sticky`	Enables macros to remain active on the interface after a link-down event.
Step 4	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.
Step 5	**show running-config interface** *interface-id* **Example:** `Switch# show running-config interface gigabit ethernet 2/0/1`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

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** **Example:** `Switch# configure terminal`	Enters global configuration mode.

Step 2	**macro auto execute** *event trigger* **builtin** *built-in macro name* [*parameter=value*] [*parameter=value*]
	
	Example:
	`Switch(config)# macro auto execute CISCO_PHONE_EVENT builtin CISCO_PHONE_AUTO_SMARTPORT ACCESS_VLAN=10 VOICE_VLAN=20`
		Defines mapping from an event trigger to a built-in macro.
		
		Specify an *event trigger*:
		• CISCO_DMP_EVENT
		• CISCO_IP_CAMERA_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** *built-in macro name*:
		
		Enter new values in the form of *name value pair* separated by spaces: [<*name1*>=<*value1*> <*name2*>=<*value2*>...]. 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_IP_CAMERA_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*
	
	Example:
	`Switch(config)# remote`
	`nvram://user:password@/C/macros`
		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:
		nvram://[[*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** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** `Switch# show running-config`	Verifies that Auto Smartports is enabled.

Step 6	**copy running-config startup-config**
	
	Example:
	`Switch# copy running-config`
	`startup-config`
		(Optional) 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:

Note   The modified value takes effect on the next link-up event or if the auto smartport is enabled after configuring the value. If the macro is already applied on an interface, then the modified value does not take effect for that 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
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
 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:

1. Configuring the remote macro in the macro.txt file.

2. Using the **macro auto execute** configuration command to specify the remote location for the macro file. In this step, note that the macro content is downloaded everytime there is a link up or link down event. Any changes made after the macro is applied or removed is reflected in the next event (macro application or removal).

Macro.txt file

```
if [[ $LINKUP == 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
```

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

In the following step you must enter the correct IP address of the remote location.

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

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

This example shows how to map a 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
```

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** **Example:** Switch# configure terminal	Enters global configuration mode.
Step 2	**shell trigger** *identifier description* **Example:** Switch(config)# shell trigger RADIUS_MAB_EVENT MAC_AuthBypass Event	Specifies the event trigger identifier and description. The identifier should have no spaces or hyphens between words.
Step 3	**end** **Example:** Switch(config)# end	Returns to privileged EXEC mode.
Step 4	**show shell triggers** **Example:** Switch# show shell triggers	Displays the event triggers on the switch.
Step 5	**copy running-config startup-config** **Example:** Switch# copy running-config startup-config	(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_AP AUTO_SMARTPORT, to replace the default VLAN with VLAN 10, and how to verify the entries.

1. Connect the device to a MAB-enabled switch port.

2. On the RADIUS server, set the attribute-value pair to **auto-smart-port**=RADIUS_MAB_EVENT.

3. On the switch, create the event trigger RADIUS_MAB_EVENT.

4. 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_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# term shell
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# term shell
Switch# show shell triggers
User defined triggers
--------------------
Built-in triggers
----------------
Trigger Id: CISCO_DMP_EVENT
Trigger namespace: ASP_TRIG
Trigger description: Digital media-player device event to apply port configuration
Trigger mapping function: CISCO_DMP_AUTO_SMARTPORT

Trigger Id: CISCO_IP_CAMERA_EVENT
Trigger namespace: ASP_TRIG
Trigger description: IP-camera device event to apply port configuration
Trigger mapping function: CISCO_IP_CAMERA_AUTO_SMARTPORT

Trigger Id: CISCO_PHONE_EVENT
Trigger namespace: ASP_TRIG
Trigger description: IP-phone device event to apply port configuration
Trigger mapping function: CISCO_PHONE_AUTO_SMARTPORT

Trigger Id: CISCO_ROUTER_EVENT
Trigger namespace: ASP_TRIG
Trigger description: Router device event to apply port configuration
Trigger mapping function: CISCO_ROUTER_AUTO_SMARTPORT

Trigger Id: CISCO_SWITCH_EVENT
Trigger namespace: ASP_TRIG
Trigger description: Switch device event to apply port configuration
Trigger mapping function: CISCO_SWITCH_AUTO_SMARTPORT

Trigger Id: CISCO_WIRELESS_AP_EVENT
Trigger namespace: ASP_TRIG
Trigger description: Autonomous ap device event to apply port configuration
Trigger mapping function: CISCO_AP_AUTO_SMARTPORT

Trigger Id: CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT
Trigger namespace: ASP_TRIG
Trigger description: Lightweight-ap device event to apply port configuration
Trigger mapping function: CISCO_LWAP_AUTO_SMARTPORT
Switch#
```

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

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

#Built-in functions:
function CISCO_AP_AUTO_SMARTPORT () {
    if [[ $LINKUP == YES ]]; then
        conf t
            interface  $INTERFACE
                macro description $TRIGGER
                switchport trunk encapsulation dot1q
                switchport trunk native vlan $NATIVE_VLAN
                switchport trunk allowed vlan ALL
                switchport mode trunk
                switchport nonegotiate
                auto qos voip trust
                mls qos trust cos
                if [[ $LIMIT == 0 ]]; then
                  default srr-queue bandwidth limit
                else
                  srr-queue bandwidth limit $LIMIT
                fi
                if [[ $SW_POE == YES ]]; then
                  if [[ $AP125X == AP125X ]]; then
                     macro description AP125X
                     macro auto port sticky
                     power inline port maximum 20000
                  fi
                fi
             exit
        end
    fi
    if [[ $LINKUP == NO ]]; then
        conf t
            interface  $INTERFACE
                no macro description
                no switchport nonegotiate
                no switchport trunk native vlan $NATIVE_VLAN
                no switchport trunk allowed vlan ALL
                no auto qos voip trust
                no mls qos trust cos
                default srr-queue bandwidth limit
                if [[ $AUTH_ENABLED == NO ]]; then
                  no switchport mode
                  no switchport trunk encapsulation
                fi
                if [[ $STICKY == YES ]]; then
                  if [[ $SW_POE == YES ]]; then
                     if [[ $AP125X == AP125X ]]; then
                        no macro auto port sticky
                        no power inline port maximum
                     fi
                  fi
                fi
            exit
        end
    fi
}
<output truncated>
```

Auto Smartports Configuration Guide

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 == 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** **Example:** `Switch# configure terminal`	Enters global configuration mode.

Step 2	**macro auto execute** *event trigger* [*parameter=value*] {*function contents*}
	
	Example:
	`Switch(config)# macro auto`
	`execute DMP_EVENT {`
	`if [[ $LINKUP == YES ]]; then`
	`conf t`
	` interface $INTERFACE`
	`   macro description $TRIGGER`
	`   switchport access vlan 1`
	`   switchport mode access`
	`   switchport port-security`
	`   switchport port-security`
	`maximum 1`
	`   switchport port-security`
	`violation restrict`
	`   switchport port-security`
	`aging time 2`
	`   switchport port-security`
	`aging type inactivity`
	`   spanning-tree portfast`
	`   spanning-tree bpduguard`
	`enable`
	`   exit`
	`fi`
	`if [[ $LINKUP == NO ]]; then`
	`conf t`
	`interface $INTERFACE`
	`    no macro description`
	`    no switchport access vlan 1`
	`    if [[ $AUTH_ENABLED == NO`
	`]]; then`
	`       no switchport mode`
	`access`
	`    fi`
	`    no switchport port-security`
	`    no switchport port-security`
	`maximum 1`
	`    no switchport port-security`
	`violation restrict`
	`    no switchport port-security`
	`aging time 2`
	`    no switchport port-security`
	`aging type inactivity`
	`    no spanning-tree portfast`
	`    no spanning-tree bpduguard`
	`enable`
	`    exit`
	`fi`
	`}`
		Specifies a user-defined macro that maps to an event trigger.
		
		{*function contents*} 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) *parameter=value*—Replace default values that begin with **$**, and enter new values in the form of name value pair separated by spaces: [*<name1>=<value1> <name2>=<value2>...*].

	Command	Purpose
Step 3	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	**show running-config** **Example:** `Switch# show running-config`	Verifies that Auto Smartports is enabled.

Step 5	**copy running-config startup-config**
	
	Example:
	`Switch# 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.

1. Connect the media player to an 802.1x- or MAB-enabled switch port.

2. On the RADIUS server, set the attribute-value pair to **auto-smart-port** =DMP_EVENT.

3. On the switch, create the event trigger DMP_EVENT, and enter the user-defined macro commands in the CLI example.

4. The switch recognizes the attribute-value pair=DMP_EVENT response from the RADIUS server and applies the macro associated with this event trigger.

Note While making the mapping changes enter the space and the semicolon exactly the same way as mentioned in the following example.

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

```
        if [[ $AUTH_ENABLED == NO ]]; then
           no switchport mode access
        fi
        no switchport port-security
        no switchport port-security maximum 1
        no switchport port-security violation restrict
        no switchport port-security aging time 2
        no switchport port-security aging type inactivity
        no spanning-tree portfast
        no spanning-tree bpduguard enable
        exit
 fi
 }
 Switch(config)# end
```

Example: User Defined Macro Triggered by MAC Address

This example shows how to create your own auto smartport macro and then bind a MAC address to that macro.

```
macro auto mac-address-group TRIGGER_PRINTER_MAC
oui list 0001E6

macro auto execute TRIGGER_PRINTER_MAC ACCESS_VLAN=10 {
if [[ $LINKUP == YES ]]
  then conf t
  interface  $INTERFACE
  macro description $TRIGGER
  switchport access vlan $ACCESS_VLAN
  switchport mode access
  switchport block unicast
  spanning-tree portfast
  switchport port-security
  switchport port-security maximum 1
  switchport port-security violation protect
  spanning-tree bpduguard enable
  exit
  end
 fi
 if [[ $LINKUP == NO ]]
  then conf t
  interface  $INTERFACE
  no macro description
  no switchport access vlan $ACCESS_VLAN
  no switchport block unicast
  no switchport port-security
  no switchport port-security maximum 1
  no switchport port-security violation protect
  no spanning-tree portfast
  no spanning-tree bpduguard enable
  exit
  end
 fi
}
```

E

Table 3-2   Supported Cisco IOS Shell Keywords

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

Table 3-3   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.

Default Static Smartports Configuration

There are no static Smartports macros enabled on the switch.

Table 3-4   *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.

NEAT Configuration

The Network Edge Access Topology (NEAT) feature extends identity to areas outside the wiring closet (such as conference rooms).

In a NEAT scenario, when 802.1x authentication is successful and an ASP macro is sent from the Access Control Server (ACS) to the switch, you must make one of the following configurations:

- Change the host mode to multi-host.

- Enable trunk configuration on the authenticator switch by configuring the cisco-av-pair as *device-traffic-class=switch* at the ACS.

Applying Static Smartports Macros

Beginning in privileged EXEC mode:

	Command	Purpose
Step 1	**show parser macro** **Example:** `Switch# show parser macro`	Displays the Cisco-default static Smartports macros embedded in the switch software.
Step 2	**show parser macro name** *macro-name* **Example:** `Switch# show parser macro name cisco-desktop`	Displays the specific macro that you want to apply.
Step 3	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 4	**macro global** {**apply**	**trace**} *macro-name* [**parameter** {*value*}] [**parameter** {*value*}] [**parameter** {*value*}] **Example:** `Switch(config)# macro global apply cisco-desktop $access_vlan 25`	Applies a macro on the switch: • To only apply each individual macro command, use the **macro global apply** *macro-name* command. • To apply and then debug a macro to find any syntax or configuration errors, use the **macro global trace** *macro-name* command. Append the macro with the required values by using the **parameter** *value* keywords. Keywords that begin with **$** require a unique parameter value. You can use the **macro global apply** *macro-name* **?** 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** *interface-id* **Example:** `Switch(config)# interface gigabitethernet 2/0/5`	(Optional) Specifies an interface and enters interface configuration mode.

	Command	Purpose
Step 6	**default interface** *interface-id*	(Optional) Clears all configuration from the specified interface.
	Example: `Switch(config)# default interface 2/1/4`	
Step 7	**macro** {**apply** | **trace**} *macro-name* [**parameter** {*value*}] [**parameter** {*value*}] [**parameter** {*value*}]	Applies a macro on the interface: • To only apply each individual macro command, use the **macro apply** *macro-name* command. • To apply and then debug a macro to find any syntax or configuration errors, use the **macro trace** *macro-name* command. Append the macro with the required values by using the **parameter** *value* keywords. Keywords that begin with **$** require a unique parameter value.
	Example: `Switch(config-if)# macro apply cisco-desktop $access_vlan 25`	You can use the **macro global apply** *macro-name* **?** 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 8	**end**	Returns to privileged EXEC mode.
	Example: `Switch(config)# end`	
Step 9	**show running-config interface** *interface-id*	Verifies that Auto Smartports is enabled.
	Example: `Switch# show running-config interface gigabit ethernet 1/0/4`	
Step 10	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.
	Example: `Switch# copy running-config startup-config`	

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# show 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 3-5   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.
show parser macro name *macro-name*	Displays a specific static Smartports macro.
show parser macro brief	Displays the static Smartports macro names.

Table 3-5   *Commands for Displaying Auto Smartports and Static Smartports Macros (continued)*

Command	Purpose
show parser macro description [**interface** *interface-id*]	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 *fetch* • environment: Displays shell environment information • functions: Displays shell functions information • triggers: Displays shell triggers information **Note** The **show shell** command is a feature at the Cisco IOS level. You may first have to enable Cisco IOS Shell by entering the **terminal shell** command before you can enter the **show shell** command. For more information, see the Cisco IOS Shell configuration guide on Cisco.com: http://www.cisco.com/en/US/docs/ios/netmgmt/configuration/guide/nm_ios_shell.pdf