CHAPTER 17

Configuring Auto SmartPort Macros

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

This chapter includes the following major sections:

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

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

If the command is not found in the Catalyst 4500 Command Reference, it is located in the larger Cisco IOS library. Refer to the *Cisco IOS Command Reference* and related publications at this location:

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

About Auto SmartPorts

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

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

Note Although Auto SmartPort detects the Cisco switch it does not invoke the event trigger automatically. The event trigger needs to be manually invoked to map the switch to macros.

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

System built-in event triggers exist for various devices based mostly on CDP and LLDP messages (Table 17-1) and some MAC address. These triggers are enabled as long as Auto SmartPort is enabled.

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

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

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

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

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

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

Configuring Auto SmartPorts

The following topics are included:

- Enabling Auto SmartPorts, page 17-3
- Auto SmartPorts Configuration Guidelines, page 17-5
- Configuring Mapping Between Event Triggers and Built-in Macros, page 17-8
- Configuring Auto SmartPorts Built-in Macro Parameters, page 17-6
- Configuring Mapping Between Event Triggers and Built-in Macros, page 17-8
- Configuring Mapping Between User-Defined Triggers and Built-in Macros, page 17-10
- Configuring Auto SmartPorts User-Defined Macros, page 17-11

Enabling Auto SmartPorts

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

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

To enable Auto SmartPorts, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# [no] macro auto global processing [fallback [cdp	lldp]]]`	Enables Auto SmartPorts on the switch globally. The fallback option allows you to enable the switch to use CDP or LLDP as a trigger when the port is configured with 802.1X and authentication fails. **Note** The fallback options only work for phone devices. Use the **no macro auto global processing fallback** command to disable the fallback mechanism, but still have Auto SmartPort enabled. Use **no macro auto global processing** to disable Auto SmartPort globally. **Note** The **macro auto processing** command turns Auto SmartPort on or off on the interface level. The default is on.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show running-config`	Verifies that Auto SmartPorts is enabled.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

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

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

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

Auto SmartPorts Default Configuration

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

Table 17-1 shows the Auto SmartPorts built-in event triggers that are embedded in the switch software by default.

Table 17-1 Auto SmartPorts Built-in Event Trigger Macros

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

Table 17-2 shows the Auto SmartPorts built-in macros that are embedded in the switch software.

Table 17-2 Auto SmartPorts Built-in Macros

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

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

Auto SmartPorts Configuration Guidelines

Auto SmartPort guidelines include the following:

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

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

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

- A port should not be a member of an EtherChannel when applying Auto SmartPorts macros.

If Auto SmartPort is not yet enabled globally, disable Auto SmartPort on all the EtherChannel ports before enabling it globally. If Auto SmartPort is already enabled, shut down the port and disable it before adding the port to an EtherChannel.

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

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

the macro (for example, auto-QoS) might conflict with an EtherChannel configuration. We recommend that you disable Auto SmartPort on interfaces belonging to EtherChannels before you enable Auto SmartPort globally. If Auto SmartPort is already enabled, disable Auto SmartPort on the interfaces before configuring EtherChannel.

- When a Cisco switch is detected on the Auto Smartport, you have to manually map the event trigger to either a built-in macro or user-defined macro. You need to also match the event trigger to the device PID.

Configuring Auto SmartPorts Built-in Macro Parameters

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

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

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 2	Switch(config)# **macro auto execute** *event trigger* **builtin** *built-in macro name* [*parameter=value*] [*parameter=value*]	Defines mapping from an event trigger to a built-in macro. Specify an *event trigger* value: • CISCO_PHONE_EVENT • CISCO_SWITCH_EVENT • CISCO_ROUTER_EVENT • CISCO_WIRELESS_AP_EVENT • CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT • CISCO_DMP_EVENT • CISCO_IPVSC_EVENT • WORD—Apply a user-defined event trigger. Specify a *built-in macro name* value: • CISCO_PHONE_AUTO_SMARTPORT (Optional) Specify the parameter values: $ACCESS_VLAN=(1) and $VOICE_VLAN=(2). • CISCO_SWITCH_AUTO_SMARTPORT (Optional) Specify the parameter values: $NATIVE_VLAN=(1). • CISCO_ROUTER_AUTO_SMARTPORT (Optional) Specify the parameter values: $NATIVE_VLAN=(1). • CISCO_AP_AUTO_SMARTPORT (Optional) Specify the parameter values: $NATIVE_VLAN=(1). • CISCO_LWAP_AUTO_SMARTPORT (Optional) Specify the parameter values: $ACCESS_VLAN=(1). • CISCO_DMP_AUTO_SMARTPORT • CISCO_IP_CAMERA_AUTO_SMARTPORT (Optional) *parameter=value*—Replace default values that begin with $. Enter new values in the form of name value pair separated by a space: [*name1=value1 name2=value2*...]. Default values are shown in parenthesis.
Step 3	Switch(config)# **end**	Returns to privileged EXEC mode.
Step 4	Switch# **show running-config**	Verifies your entries.
Step 5	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

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

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

```
Switch# configure terminal
Switch(config)# macro auto execute CISCO_PHONE_EVENT builtin CISCO_PHONE_AUTO_SMARTPORT
ACCESS_VLAN=10 VOICE_VLAN=20
Switch(config)#
Switch(config)#
Switch(config)#!!! the next command enables auto smart ports globally
Switch(config)# macro auto global processing fallback cdp
Switch(config)#
Switch(config)# exit

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

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

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

Configuring Mapping Between Event Triggers and Built-in Macros

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

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

	Command	Purpose
Step 1	Switch# **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
```

Configuring User-Defined Event Triggers

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

The following sections describe these triggers:

- 802.1X-Based Event Trigger, page 17-9
- MAC Address-Based Event Trigger, page 17-10

802.1X-Based Event Trigger

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

To configure an event trigger, perform this task:

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

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

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

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

MAC Address-Based Event Trigger

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

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

Use the **no macro auto mac-address-group** *grp_name* to delete the event trigger.

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

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

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

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`macro auto execute`** *`event trigger`* **`builtin`** *`built-in macro name`* `[`*`parameter=value`*`] [`*`parameter=value`*`]`	Specifies a user-defined event trigger and a macro name. This action replaces built-in macro default values, and configures mapping from an event trigger to a built-in Auto Smartports macro. **Note** When performing a mapping, you must provide parameter values. For example, you must specify $ACCESS_VLAN=(1) and $VOICE_VLAN=(2) for the macro CISCO_PHONE_AUTO_SMARTPORT.
Step 3	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 4	`Switch#` **`show shell triggers`**	Displays the event triggers on the switch.
Step 5	`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_PHONE_AUTO_SMARTPORT with access VLAN set to 10, and how to verify the entries.

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

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

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

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

The switch recognizes the attribute-value pair=RADIUS_MAB_EVENT response from the RADIUS server and applies the macro CISCO_PHONE_AUTO_SMARTPORT, as in the following example:

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

Configuring Auto SmartPorts User-Defined Macros

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

Inside a user-defined macro, besides parameters specified through **macro auto execute trigger parameter-name=value ..,** you also can use the following variables published by EEM (Table 17-3).

Table 17-3 Variables Published by EEM

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

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

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`macro auto execute`** *`event trigger`* `[`*`parameter=value`*`] {`*`function contents`*`}`	Specifies a user-defined macro that maps to an event trigger. Specify an *event trigger* value: • CISCO_PHONE_EVENT • CISCO_SWITCH_EVENT • CISCO_ROUTER_EVENT • CISCO_WIRELESS_AP_EVENT • CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT • WORD Applies a user-defined event trigger. • CISCO_DMP_EVENT • CISCO_IPVSC_EVENT *function contents*—Specifies a user-defined macro to associate with the trigger. Enter the macro contents within braces. Begin the Cisco IOS shell commands with the left brace and end the command grouping with the right brace. (Optional) *parameter*=*value*—Replaces default values that begin with **$**, enter new values in the form of name value pair separated by a space: [*name1*=***value1*** *name2*=*value2*...].
Step 3	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 4	`Switch#` **`show running-config`**	Verifies your entries.
Step 5	`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 Cisco digital media player (DMP) to a user-defined macro.

Step 1 Connect the DMP to an 802.1X- or MAB-enabled switch port.

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

Step 3 On the switch, create the event trigger CISCO_DMP_EVENT, and map it to the user-defined macro commands shown below.

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

The following example shows the macro portion of the automacro:

```
Switch(config)# shell trigger CISCO_DMP_EVENT Cisco DMP player
Switch(config)# macro auto execute CISCO_DMP_EVENT {
if [[ $LINKUP -eq YES ]]; then
conf t
 interface $INTERFACE
   macro description $TRIGGER
   switchport access vlan 1
   switchport mode access
   switchport port-security
   switchport port-security maximum 1
   switchport port-security violation restrict
   switchport port-security aging time 2
   switchport port-security aging type inactivity
   spanning-tree portfast
   spanning-tree bpduguard enable
   exit
fi
```

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

```
if [[ $LINKUP -eq NO ]]; then
conf t
interface $INTERFACE
     no macro description $TRIGGER
     no switchport access vlan 1
     if [[ $AUTH_ENABLED -eq NO ]]; then
        no switchport mode access
     fi
     no switchport port-security
     no switchport port-security maximum 1
     no switchport port-security violation restrict
     no switchport port-security aging time 2
     no switchport port-security aging type inactivity
     no spanning-tree portfast
     no spanning-tree bpduguard enable
     exit
fi
}
Switch(config)# end
```

Table 17-4 lists the supported shell keywords your can apply in your macros and antimacro statements.

Table 17-4 Supported Cisco IOS Shell Keywords

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

Table 17-5 lists the shell keywords that are not supported in macros and antimacros.

Table 17-5 Unsupported Cisco IOS Shell Reserved Keywords

Command	Description
|	Pipeline.
case	Conditional construct.
esac	Conditional construct.
for	Looping construct.
function	Shell function.
in	Conditional construct.
select	Conditional construct.
time	Pipeline.
until	Looping construct.
while	Looping construct.

Displaying Auto SmartPorts

To display the Auto SmartPorts and static SmartPorts macros, use one or more of the privileged EXEC commands in Table 17-6.

Table 17-6 Commands for Displaying Auto SmartPorts and Static SmartPorts Macros

Command	Purpose
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.
show parser macro description [**interface** *interface-id*]	Displays the static SmartPorts macro description for all interfaces or for a specified interface.
show shell [**triggers** | **functions**]	Displays information about Auto SmartPorts event triggers and macros.

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

```
Switch# show shell triggers

User defined triggers
---------------------
Built-in triggers
-----------------
Trigger Id: CISCO_PHONE_EVENT
Trigger description: Event for ip-phone macro
Trigger environment: ACCESS_VLAN=1 VOICE_VLAN=2
Trigger mapping function: CISCO_PHONE_AUTO_SMARTPORT

Trigger Id: CISCO_ROUTER_EVENT
Trigger description: Event for router macro
Trigger environment: NATIVE_VLAN=1
Trigger mapping function: CISCO_ROUTER_AUTO_SMARTPORT

Trigger Id: CISCO_SWITCH_EVENT
Trigger description: Event for switch macro
Trigger environment: NATIVE_VLAN=1
Trigger mapping function: CISCO_SWITCH_AUTO_SMARTPORT

Trigger Id: CISCO_WIRELESS_AP_EVENT
Trigger description: Event for Wireless Access Point macro
Trigger environment: NATIVE_VLAN=1
Trigger mapping function: CISCO_AP_AUTO_SMARTPORT

Trigger Id: CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT
Trigger description: Event for Wireless Lightweight Access Point macro
Trigger environment: NATIVE_VLAN=1
Trigger mapping function: CISCO_LWAP_AUTO_SMARTPORT
```

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

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

#Built-in functions:
function CISCO_AP_AUTO_SMARTPORT () {
    if [[ $LINKUP -eq YES ]]; then
        conf t
            interface  $INTERFACE
                macro description $TRIGGER
                switchport trunk encapsulation dot1q
                switchport trunk native vlan $NATIVE_VLAN
                switchport trunk allowed vlan ALL
                switchport mode trunk
                switchport nonegotiate
                auto qos voip trust
                mls qos trust cos
            exit
        end
    fi
    if [[ $LINKUP -eq NO ]]; then
        conf t
            interface  $INTERFACE
                no macro description
                no switchport nonegotiate
                no switchport trunk native vlan $NATIVE_VLAN
                no switchport trunk allowed vlan ALL
                no auto qos voip trust
                no mls qos trust cos
                if [[ $AUTH_ENABLED -eq NO ]]; then
                    no switchport mode
                    no switchport trunk encapsulation
                fi
            exit
        end
    fi
}

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

<output truncated>
```