Preface

Document Conventions

This document uses the following conventions:

Convention	Description
^ or Ctrl	Both the ^ symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination **^D** or **Ctrl-D** means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
`courier` font	Terminal sessions and information the system displays appear in `courier` font.
`Bold courier` font	Bold courier font indicates text that the user must enter.
[x]	Elements in square brackets are optional.
...	An ellipsis (three consecutive nonbolded periods without spaces) after a syntax element indicates that the element can be repeated.
|	A vertical line, called a pipe, indicates a choice within a set of keywords or arguments.
{x | y}	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

`courier` font	Terminal sessions and information the system displays appear in `courier` font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Reader Alert Conventions

This document uses the following conventions for reader alerts:

Note Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

Tip Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Warning **Means *reader be warned*. In this situation, you might perform an action that could result in bodily injury.**

Related Documentation

Note Before installing or upgrading the switch, refer to the switch release notes.

- Catalyst 3650 Switch documentation at:

  http://www.cisco.com/go/cat3650_docs
- Cisco SFP and SFP+ modules documentation, including compatibility matrixes at:

  http://www.cisco.com/en/US/products/hw/modules/ps5455/tsd_products_support_series_home.html
- Error Message Decoder, located at:

  https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed, and set content to be delivered directly to your desktop by a reader application. The RSS feeds are a free service, and Cisco supports RSS version 2.0.

Using the Command-Line Interface

This chapter describes the Cisco IOS command-line interface (CLI) and how to use it to configure your standalone switch or a switch stack, referred to as the *switch*. It contains these sections:

Understanding Command Modes

The Cisco IOS user interface is divided into many different modes. The commands available to you depend on which mode you are currently in. Enter a question mark (?) at the system prompt to obtain a list of commands available for each command mode.

When you start a session on the switch, you begin in user mode, often called user EXEC mode. Only a limited subset of the commands are available in user EXEC mode. For example, most of the user EXEC commands are one-time commands, such as **show** commands, which show the current configuration status, and **clear** commands, which clear counters or interfaces. The user EXEC commands are not saved when the switch reboots.

To have access to all commands, you must enter privileged EXEC mode. Normally, you must enter a password to enter privileged EXEC mode. From this mode, you can enter any privileged EXEC command or enter global configuration mode.

Using the configuration modes (global, interface, and line), you can make changes to the running configuration. If you save the configuration, these commands are stored and used when the switch reboots. To access the various configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and line configuration mode.

Table 1-1 describes the main command modes, how to access each one, the prompt you see in that mode, and how to exit the mode. The examples in the table use the hostname *Switch*.

Table 1-1 Command Mode Summary

Mode	Access Method	Prompt	Exit Method	About This Mode
User EXEC	Begin a session with your switch.	`Switch>`	Enter **logout** or **quit**.	Use this mode to • Change terminal settings. • Perform basic tests. • Display system information.
Privileged EXEC	While in user EXEC mode, enter the **enable** command.	`Switch#`	Enter **disable** to exit.	Use this mode to verify commands that you have entered. Use a password to protect access to this mode.
Global configuration	While in privileged EXEC mode, enter the **configure** command.	`Switch(config)#`	To exit to privileged EXEC mode, enter **exit** or **end**, or press **Ctrl-Z**.	Use this mode to configure parameters that apply to the entire switch.
VLAN configuration	While in global configuration mode, enter the **vlan** *vlan-id* command.	`Switch(config-vlan)#`	To exit to global configuration mode, enter the **exit** command. To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**.	Use this mode to configure VLAN parameters. When VTP mode is transparent, you can create extended-range VLANs (VLAN IDs greater than 1005) and save configurations in the switch startup configuration file.
Interface configuration	While in global configuration mode, enter the **interface** command (with a specific interface).	`Switch(config-if)#`	To exit to global configuration mode, enter **exit**. To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**.	Use this mode to configure parameters for the Ethernet ports.
Line configuration	While in global configuration mode, specify a line with the **line vty** or **line console** command.	`Switch(config-line)#`	To exit to global configuration mode, enter **exit**. To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**.	Use this mode to configure parameters for the terminal line.

Understanding the Help System

You can enter a question mark (?) at the system prompt to display a list of commands available for each command mode. You can also obtain a list of associated keywords and arguments for any command, as shown in Table 1-2.

Table 1-2 Help Summary

Command	Purpose
help	Obtain a brief description of the help system in any command mode.

*abbreviated-command-entry***?**	Obtain a list of commands that begin with a particular character string.
	For example:
	`Switch# di?`
	`dir disable disconnect`

abbreviated-command-entry<**Tab**>	Complete a partial command name.
	For example:
	`Switch# sh conf<tab>`
	`Switch# show configuration`

?	List all commands available for a particular command mode. For example: `Switch> ?`
command **?**	List the associated keywords for a command. For example: `Switch> show ?`

command keyword **?**	List the associated arguments for a keyword.
	For example:
	`Switch(config)# cdp holdtime ?`
	`  <10-255> Length of time (in sec) that receiver must keep this packet`

Understanding Abbreviated Commands

You need to enter only enough characters for the switch to recognize the command as unique.

This example shows how to enter the **show configuration** privileged EXEC command in an abbreviated form:

```
Switch# show conf
```

Understanding no and default Forms of Commands

Almost every configuration command also has a **no** form. In general, use the **no** form to disable a feature or function or reverse the action of a command. For example, the **no shutdown** interface configuration command reverses the shutdown of an interface. Use the command without the keyword **no** to re-enable a disabled feature or to enable a feature that is disabled by default.

Configuration commands can also have a **default** form. The **default** form of a command returns the command setting to its default. Most commands are disabled by default, so the **default** form is the same as the **no** form. However, some commands are enabled by default and have variables set to certain default values. In these cases, the **default** command enables the command and sets variables to their default values.

Understanding CLI Error Messages

Table 1-3 lists some error messages that you might encounter while using the CLI to configure your switch.

Table 1-3 Common CLI Error Messages

Error Message	Meaning	How to Get Help
`% Ambiguous command: "show con"`	You did not enter enough characters for your switch to recognize the command.	Re-enter the command followed by a question mark (?) with a space between the command and the question mark. The possible keywords that you can enter with the command appear.
`% Incomplete command.`	You did not enter all the keywords or values required by this command.	Re-enter the command followed by a question mark (?) with a space between the command and the question mark. The possible keywords that you can enter with the command appear.
`% Invalid input detected at '^' marker.`	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode. The possible keywords that you can enter with the command appear.

Using Configuration Logging

You can log and view changes to the switch configuration. You can use the Configuration Change Logging and Notification feature to track changes on a per-session and per-user basis. The logger tracks each configuration command that is applied, the user who entered the command, the time that the command was entered, and the parser return code for the command. This feature includes a mechanism for asynchronous notification to registered applications whenever the configuration changes. You can choose to have the notifications sent to the syslog.

Note Only CLI or HTTP changes are logged.

Using Command History

The software provides a history or record of commands that you have entered. The command history feature is particularly useful for recalling long or complex commands or entries, including access lists. You can customize this feature to suit your needs as described in these sections:

- Changing the Command History Buffer Size, page 1-5 (optional)
- Recalling Commands, page 1-5 (optional)
- Disabling the Command History Feature, page 1-6 (optional)

Changing the Command History Buffer Size

By default, the switch records ten command lines in its history buffer. You can alter this number for a current terminal session or for all sessions on a particular line. These procedures are optional.

Beginning in privileged EXEC mode, enter this command to change the number of command lines that the switch records during the current terminal session:

```
Switch# terminal history [size number-of-lines]
```

The range is from 0 to 256.

Beginning in line configuration mode, enter this command to configure the number of command lines the switch records for all sessions on a particular line:

```
Switch(config-line)# history [size number-of-lines]
```

The range is from 0 to 256.

Recalling Commands

To recall commands from the history buffer, perform one of the actions listed in Table 1-4. These actions are optional.

Table 1-4 Recalling Commands

Action[1]	Result
Press **Ctrl-P** or the up arrow key.	Recall commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Press **Ctrl-N** or the down arrow key.	Return to more recent commands in the history buffer after recalling commands with **Ctrl-P** or the up arrow key. Repeat the key sequence to recall successively more recent commands.
show history	While in privileged EXEC mode, list the last several commands that you just entered. The number of commands that appear is controlled by the setting of the **terminal history** global configuration command and the **history** line configuration command.

[1] The arrow keys function only on ANSI-compatible terminals such as VT100s.

Disabling the Command History Feature

The command history feature is automatically enabled. You can disable it for the current terminal session or for the command line. These procedures are optional.

To disable the feature during the current terminal session, enter the **terminal no history** privileged EXEC command.

To disable command history for the line, enter the **no history** line configuration command.

Using Editing Features

This section describes the editing features that can help you manipulate the command line. It contains these sections:

- Enabling and Disabling Editing Features, page 1-6 (optional)
- Editing Commands through Keystrokes, page 1-6 (optional)
- Editing Command Lines that Wrap, page 1-8 (optional)

Enabling and Disabling Editing Features

Although enhanced editing mode is automatically enabled, you can disable it, re-enable it, or configure a specific line to have enhanced editing. These procedures are optional.

To globally disable enhanced editing mode, enter this command in line configuration mode:

```
Switch (config-line)# no editing
```

To re-enable the enhanced editing mode for the current terminal session, enter this command in privileged EXEC mode:

```
Switch# terminal editing
```

To reconfigure a specific line to have enhanced editing mode, enter this command in line configuration mode:

```
Switch(config-line)# editing
```

Editing Commands through Keystrokes

Table 1-5 shows the keystrokes that you need to edit command lines. These keystrokes are optional.

Table 1-5 Editing Commands through Keystrokes

Capability	Keystroke[1]	Purpose
Move around the command line to make changes or corrections.	Press **Ctrl-B**, or press the left arrow key.	Move the cursor back one character.

Table 1-5 Editing Commands through Keystrokes (continued)

Capability	Keystroke[1]	Purpose
	Press **Ctrl-F**, or press the right arrow key.	Move the cursor forward one character.
	Press **Ctrl-A**.	Move the cursor to the beginning of the command line.
	Press **Ctrl-E**.	Move the cursor to the end of the command line.
	Press **Esc B**.	Move the cursor back one word.
	Press **Esc F**.	Move the cursor forward one word.
	Press **Ctrl-T**.	Transpose the character to the left of the cursor with the character located at the cursor.
Recall commands from the buffer and paste them in the command line. The switch provides a buffer with the last ten items that you deleted.	Press **Ctrl-Y.**	Recall the most recent entry in the buffer.
	Press **Esc Y**.	Recall the next buffer entry. The buffer contains only the last 10 items that you have deleted or cut. If you press **Esc Y** more than ten times, you cycle to the first buffer entry.
Delete entries if you make a mistake or change your mind.	Press the **Delete** or **Backspace** key.	Erase the character to the left of the cursor.
	Press **Ctrl-D**.	Delete the character at the cursor.
	Press **Ctrl-K**.	Delete all characters from the cursor to the end of the command line.
	Press **Ctrl-U** or **Ctrl-X**.	Delete all characters from the cursor to the beginning of the command line.
	Press **Ctrl-W**.	Delete the word to the left of the cursor.
	Press **Esc D**.	Delete from the cursor to the end of the word.
Capitalize or lowercase words or capitalize a set of letters.	Press **Esc C**.	Capitalize at the cursor.
	Press **Esc L**.	Change the word at the cursor to lowercase.
	Press **Esc U**.	Capitalize letters from the cursor to the end of the word.
Designate a particular keystroke as an executable command, perhaps as a shortcut.	Press **Ctrl-V** or **Esc Q**.	

Table 1-5 Editing Commands through Keystrokes (continued)

Capability	Keystroke[1]	Purpose
Scroll down a line or screen on displays that are longer than the terminal screen can display. **Note** The More prompt is used for any output that has more lines than can be displayed on the terminal screen, including **show** command output. You can use the **Return** and **Space** bar keystrokes whenever you see the More prompt.	Press the **Return** key.	Scroll down one line.
	Press the **Space** bar.	Scroll down one screen.
Redisplay the current command line if the switch suddenly sends a message to your screen.	Press **Ctrl-L** or **Ctrl-R.**	Redisplay the current command line.

[1] The arrow keys function only on ANSI-compatible terminals such as VT100s.

Editing Command Lines that Wrap

You can use a wraparound feature for commands that extend beyond a single line on the screen. When the cursor reaches the right margin, the command line shifts ten spaces to the left. You cannot see the first ten characters of the line, but you can scroll back and check the syntax at the beginning of the command. The keystroke actions are optional.

To scroll back to the beginning of the command entry, press **Ctrl-B** or the left arrow key repeatedly. You can also press **Ctrl-A** to immediately move to the beginning of the line.

Note The arrow keys function only on ANSI-compatible terminals such as VT100s.

In this example, the **access-list** global configuration command entry extends beyond one line. When the cursor first reaches the end of the line, the line is shifted ten spaces to the left and redisplayed. The dollar sign ($) shows that the line has been scrolled to the left. Each time the cursor reaches the end of the line, the line is again shifted ten spaces to the left.

```
Switch(config)# access-list 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1
Switch(config)# $ 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1.20 255.25
Switch(config)# $t tcp 131.108.2.5 255.255.255.0 131.108.1.20 255.255.255.0 eq
Switch(config)# $108.2.5 255.255.255.0 131.108.1.20 255.255.255.0 eq 45
```

After you complete the entry, press **Ctrl-A** to check the complete syntax before pressing the **Return** key to execute the command. The dollar sign ($) appears at the end of the line to show that the line has been scrolled to the right:

```
Switch(config)# access-list 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1$
```

The software assumes you have a terminal screen that is 80 columns wide. If you have a width other than that, use the **terminal width** privileged EXEC command to set the width of your terminal.

Use line wrapping with the command history feature to recall and modify previous complex command entries.

Searching and Filtering Output of show and more Commands

You can search and filter the output for **show** and **more** commands. This is useful when you need to sort through large amounts of output or if you want to exclude output that you do not need to see. Using these commands is optional.

To use this functionality, enter a **show** or **more** command followed by the *pipe* character (|), one of the keywords **begin**, **include**, or **exclude**, and an expression that you want to search for or filter out:

command **|** {**begin** | **include** | **exclude**} *regular-expression*

Expressions are case sensitive. For example, if you enter **| exclude output**, the lines that contain *output* are not displayed, but the lines that contain *Output* appear.

This example shows how to include in the output display only lines where the expression *protocol* appears:

```
Switch# show interfaces | include protocol
Vlan1 is up, line protocol is up
Vlan10 is up, line protocol is down
GigabitEthernet1/0/1 is up, line protocol is down
GigabitEthernet1/0/2 is up, line protocol is up
```

Accessing the CLI

You can access the CLI through a console connection, through Telnet, or by using the browser.

You manage the switch stack and the stack member interfaces through the stack master. You cannot manage stack members on an individual switch basis. You can connect to the stack master through the console port or the Ethernet management port of one or more stack members. Be careful with using multiple CLI sessions to the stack master. Commands you enter in one session are not displayed in the other sessions. Therefore, it is possible to lose track of the session from which you entered commands.

Note We recommend using one CLI session when managing the switch stack.

If you want to configure a specific stack member port, you must include the stack member number in the CLI command interface notation.

To debug a specific stack member, you can access it from the stack master by using the **session** *stack-member-number* privileged EXEC command. The stack member number is appended to the system prompt. For example, `Switch-2#` is the prompt in privileged EXEC mode for stack member 2, and where the system prompt for the stack master is `Switch`. Only the **show** and **debug** commands are available in a CLI session to a specific stack member.

Accessing the CLI through a Console Connection or through Telnet

Before you can access the CLI, you must connect a terminal or a PC to the switch console or connect a PC to the Ethernet management port and then power on the switch, as described in the hardware installation guide that shipped with your switch.

If your switch is already configured, you can access the CLI through a local console connection or through a remote Telnet session, but your switch must first be configured for this type of access.

You can use one of these methods to establish a connection with the switch:

- Connect the switch console port to a management station or dial-up modem, or connect the Ethernet management port to a PC. For information about connecting to the console or Ethernet management port, see the switch hardware installation guide.
- Use any Telnet TCP/IP or encrypted Secure Shell (SSH) package from a remote management station. The switch must have network connectivity with the Telnet or SSH client, and the switch must have an enable secret password configured.
  - The switch supports up to 16 simultaneous Telnet sessions. Changes made by one Telnet user are reflected in all other Telnet sessions.
  - The switch supports up to five simultaneous secure SSH sessions.

After you connect through the console port, through the Ethernet management port, through a Telnet session or through an SSH session, the user EXEC prompt appears on the management station.

CHAPTER 2

Auto Smartports and Static Smartports Macros

Auto Smartports Macros

Auto Smartports 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 macro. When there is a link-down event, the switch removes the macro. For example, when you connect a Cisco IP phone to a port, Auto Smartports applies the IP phone macro. The 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 port macros.

The macros embedded in the switch software are groups of CLI commands.

You can also create user-defined macros by using the Cisco IOS Shell scripting capability, which is a BASH-like language syntax for command automation and variable replacement.

Static Smartports Macros

Static Smartports macros provide port configurations that you apply manually based on the device connected to the port. When you apply a static macro, the macro CLI commands are added to the existing port configuration. When there is a link-down event on the port, the switch does not remove the static macro configuration.

Event Triggers

Auto Smartports uses event triggers to map macros to the source port of the event. The most common triggers are based on Cisco Discovery Protocol (CDP) messages received from another device. A CDP event trigger occurs when these devices are detected:

- Cisco switch
- Cisco router
- Cisco IP Phone
- Cisco Wireless Access Points, including autonomous and lightweight access points
- Cisco IP video surveillance camera
- Cisco digital media player

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.

Additional event triggers for Cisco and third-party devices are user-defined MAC address groups, MAC authentication bypass (MAB) messages, 802.1x authentication messages, and Link Layer Discovery Protocol (LLDP) messages.

LLDP supports a set of attributes used to discover neighbor devices. These type, length, and value attributes and descriptions are referred to as TLVs. LLDP-supported devices use TLVs to receive and send information. This protocol advertises details such as device configuration information, capabilities, and identity. Auto Smartports uses the LLDP *system capabilities* TLV as the event trigger. Use the event trigger control feature to specify if the switch applies a macro based on the detection method, device type, or configured trigger.

For more information about configuring the LLDP system capabilities TLV attributes for Auto Smartports, see the “Configuring LLDP, LLDP-MED, and Wired Location Service” chapter in the switch-specific software configuration guides.

For devices that do not support CDP, LLDP, MAB, or 802.1x authentication, such as network printers, laptops, or legacy Cisco Digital Media Players, you can configure a MAC address group with a MAC operationally unique identifier (OUI)-based trigger or MAC address based trigger. You map the MAC address group to a built-in or user-defined macro that has the desired configuration.

User-Defined Files

You can designate a remote server location for user-defined macro files. You can then update and maintain one set of macro files for use by multiple switches across the network.

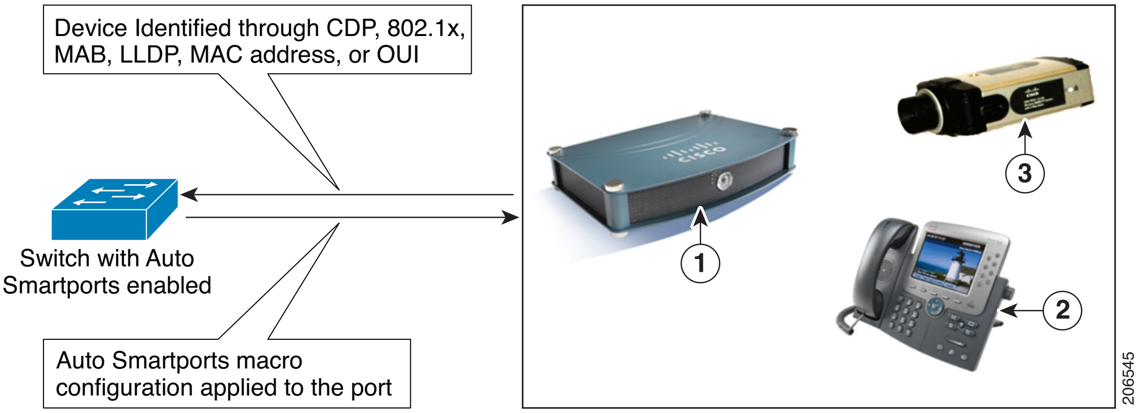
Macro Persistence

The macro persistence feature causes macro configurations to remain applied on the switch ports regardless of a link-down event. This eliminates multiple system log and configuration change notifications when the switch has link-up and link-down events or is a domain member or an end point in an EnergyWise network.

Auto Smartports and Cisco Medianet

Cisco Medianet enables intelligent services in the network infrastructure for a variety of video applications. A service of Medianet is autoprovisioning for Cisco Digital Media Players and Cisco IP video surveillance cameras through Auto Smartports. The switch identifies Cisco and third-party video devices by using CDP, 802.1x, MAB, LLDP, and MAC addresses (Figure 2-1). The switch applies the applicable macro to enable the appropriate VLAN, standard quality of service (QoS), and auto-QoS settings for the device. The switch also uses a built-in MAC address group to detect the legacy Cisco digital media player (DMP), based on an OUI of of4400 or 23ac00. You can also create custom user-defined macros for any video device.

Figure 2-1 Cisco Medianet Deployment Example


Device Identified through CDP, 802.1x, MAB, LLDP, MAC address, or OUI
Switch with Auto Smartports enabled
Auto Smartports macro configuration applied to the port
1
2
3
206545

1	Digital media player	**3**	Cisco IP video surveillance camera
2	Cisco IP phone		

CHAPTER 3

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 3650 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_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# 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_IPVSC_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>
```

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
}
```

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.

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

Command	Description
$	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.

	Command	Purpose
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.
Step 6	**default interface** *interface-id* **Example:** `Switch(config)# default interface 2/1/4`	(Optional) Clears all configuration from the specified interface.

	Command	Purpose
Step 7	**macro** {**apply** | **trace**} *macro-name* [**parameter** {*value*}] [**parameter** {*value*}] [**parameter** {*value*}] **Example:** `Switch(config-if)# macro apply cisco-desktop $access_vlan 25`	Applies a macro on the interface: • To only apply each individual macro command, use the **macro apply** *macro-name* command. • To apply and then debug a macro to find any syntax or configuration errors, use the **macro trace** *macro-name* command. Append the macro with the required values by using the **parameter** *value* keywords. Keywords that begin with **$** require a unique parameter value. 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**
	Example:
	`Switch(config)# end`
		Returns to privileged EXEC mode.

Step 9	**show running-config interface** *interface-id* **Example:** `Switch# show running-config interface gigabit ethernet 1/0/4`	Verifies that Auto Smartports is enabled.
Step 10	**copy running-config startup-config** **Example:** `Switch# copy running-config startup-config`	(Optional) Saves 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# 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.
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

CHAPTER 4

Auto Smartports and Static Smartports Macros CLI Commands

macro

To apply a macro to an interface or to apply and debug a macro on an interface, use the **macro** interface configuration command.

macro {**apply** | **trace**} *macro-name* [**parameter** {*value*}] [**parameter** {*value*}] [**parameter** {*value*}]

Syntax Description

apply	Applies a macro to an interface.
trace	Applies a macro to an interface and then debugs it.
macro-name	Specifies the name of the macro.
parameter *value*	(Optional) Specifies unique parameter values that are specific to the interface. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value.

Command Default

None.

Command Modes

Interface configuration

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

You can use the **macro apply** *macro-name* interface configuration command to apply and show the macros running on an interface.

You can use the **macro trace** *macro-name* interface configuration command to apply and then debug the macro to find any syntax or configuration errors.

If a command fails because of a syntax error or a configuration error when you apply a macro, the macro continues to apply the remaining commands to the interface.

When creating a macro that requires the assignment of unique values, use the **parameter** *value* keywords to designate values specific to the interface.

Keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value. Any full match of a keyword, even if it is part of a larger string, is considered a match and is replaced by the corresponding value.

Some macros might contain keywords that require a parameter value. You can use the **macro apply** *macro-name* **?** command to display a list of any required values in the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.

There are Cisco-default Smartports macros embedded in the switch software. You can display these macros and the commands that they contain by using the **show parser macro** user EXEC command.

Follow these guidelines when you apply a Cisco-default Smartports macro on an interface:

- Display all macros on the switch by using the **show parser macro** user EXEC command. Display the contents of a specific macro by using the **show parser macro name** *macro-name* user EXEC command.
- Keywords that begin with **$** indicate that a unique parameter value is required. Append the Cisco-default macro with the required values by using the **parameter** *value* keywords.

  The Cisco-default macros use the **$** character to identify required keywords. You can use the **$** character to define keywords when you create a macro.

When you apply a macro to an interface, the macro name is automatically added to the interface. You can display the applied commands and macro names by using the **show running-config interface** *interface-id* user EXEC command.

A macro applied to an interface range functions the same way as a macro applied 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.

You can delete a macro-applied configuration on an interface by entering the **default interface** *interface-id* interface configuration command.

Examples

After you use the **macro name** global configuration command, you can apply it to an interface. This example shows how to apply a user-created macro called *duplex* to an interface:

```
Switch(config-if)# macro apply duplex
```

To debug a macro, use the **macro trace** interface configuration command to find any syntax or configuration errors in the macro as it is applied to an interface.

```
Switch(config-if)# macro trace duplex
Applying command...'duplex auto'
%Error Unknown error.
Applying command...'speed nonegotiate'
```

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

```
Switch# show parser macro cisco-desktop
--------------------------------------------------------------
Macro name : cisco-desktop
Macro type : default

# Basic interface - Enable data VLAN only
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID
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 $AVID 25
```

Related Commands

Command	Description
macro description	Adds a description about the macros that are applied to an interface. This command is required for Auto Smartports to function.
macro global	Applies a macro on a switch or applies and traces a macro on a switch.
macro global description	Adds a description about the macros that are applied to the switch.
show parser macro	Displays the macro definition for all macros or for the specified macro.

macro auto device

To replace macro default parameter values with values that are specific to your switch, use the **macro auto device** global configuration command. Use the **no** form of this command to remove the parameter values.

macro auto device {**access-point** | **ip-camera** | **lightweight-ap** | **media-player** | **phone** | **router** | **switch**} [*parameter*=*value*]

no macro auto device {**access-point** | **ip-camera** | **lightweight-ap** | **media-player** | **phone** | **router** | **switch**} [*parameter*=*value*]

Syntax Description

access-point	Replaces the access-point default parameter value: NATIVE_VLAN=1
ip-camera	Replaces the IP video surveillance camera default parameter value: ACCESS_VLAN=1
lightweight-ap	Replaces the lightweight access point default parameter value: ACCESS_VLAN=1
media-player	Replaces the digital media player default parameter value: ACCESS_VLAN=1
phone	Replaces the IP phone default parameter values: ACCESS_VLAN=1 VOICE_VLAN=2
router	Replaces the router default parameter value: NATIVE_VLAN=1
switch	Replaces the switch default parameter value: NATIVE_VLAN=1
parameter=*value*	(Optional) Replaces the macro default parameter values. Enter new values in the form of name value pair separated by spaces: [<name1>=<value1> <name2>=<value2>...]

Command Default

Macro default parameter values are defined previously.

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

Use the **macro auto device** global configuration command to replace macro default parameter values with values that are specific to your switch. Use the **no** form of this command to remove the parameter values.

Use the **show macro device** privileged EXEC command to display the contents of the macros. Use the *parameter*=*value* keywords to replace default parameter values within a specific macro.

You can also use the **macro auto execute** global configuration command to specify default parameter values. This command also requires that you specify an event trigger and a built-in or user-defined macro. If you enable both the **macro auto device** and the **macro auto execute** commands, the parameters specified in the command last executed is applied to the switch. Only one command is active on the switch.

To verify that a macro is applied to an interface, use the **show macro auto interface** user EXEC command.

Examples

This example shows how to display the IP phone macro parameter values, enable the IP phone macro, and change the default voice VLAN to 20:

```
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
```

Related Commands

Command	Description
macro auto execute	Configures mapping from an event trigger to a built-in macro.
macro auto global processing	Enables Auto Smartports on a switch.
macro auto mac-address-group	Configures MAC address groups.
macro auto sticky	Configures macro persistence.
shell trigger	Creates event triggers.
show macro auto	Displays information about macros.
show shell	Displays information about event triggers and macros.

macro auto execute

To replace built-in macro default values and to configure mapping from an event trigger to a built-in or user-defined macro, use the **macro auto execute** global configuration command.

macro auto execute *event trigger* {[**builtin** *built-in macro name*] | [**remote** *url*]} [*parameter=value*] [*{function contents}*]

no macro auto execute *event trigger* {[**builtin** *built-in macro name*] | [**remote** *url*]} [*parameter=value*] [*{function contents}*]

Syntax Description

event trigger	Defines mapping from an event trigger to a built-in macro. Specifies 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 such as a MAC address group.
builtin *built-in macro name*	(Optional) Specifies a **builtin** *built-in macro name*: • 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.
parameter=value	(Optional) *parameter=value*—Replaces default values for parameter values shown for the *bultin-macro name,* for example, ACCESS_VLAN=1. Enter new values in the form of name value pair separated by a space: [<name1>=<value1> <name2>=<value2>...].
{function contents}	(Optional) *{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.

remote *url*	(Optional) Specifies a remote server location: • 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 the 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*

Command Default

None.

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

Use the **macro auto execute** global configuration command to replace the built-in macro default values with values that are specific to your switch.

The switch automatically maps from event triggers to built-in macros. The built-in macros are system-defined macros in the software image. You can also create user-defined macros by using the Cisco IOS shell scripting capability.

You can create new event triggers by using the **shell trigger** global configuration commands. Use the **show shell triggers** privileged EXEC command to display the contents of the user-defined triggers and macros.

You can use the **macro auto mac-address-group** global configuration command to create event triggers for devices that do not support Cisco Discovery Protocol (CDP) or Link Layer Discovery Protocol (LLDP).

You can use the remote macro feature to store macros in a central location for designated network switches to use. You can then maintain and update the macro files for use by multiple switches. Use **remote** *url* to configure the remote server location and macro path information. There are no specific file extension requirements for saved macro files.

Auto Smartports macros and antimacros (the antimacro is the portion of the applied macro that removes it at link down) have these guidelines and limitations:

- You can 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 enabling Auto Smartports on the switch.
- 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.
- The built-in-macro default data VLAN is VLAN 1. The 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.
- For 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.*
- The switch supports Auto Smartport macros only on directly connected devices. Multiple device connections, such as hubs, are not supported.
- If authentication is enabled on a port, the switch ignores a MAC address trigger if authentication fails.
- The order of CLI commands within the macro and the corresponding antimacro can be different.

Examples

This example shows how to use two built-in 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
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 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.
4. The switch recognizes the attribute-value pair=DMP_EVENT response from the RADIUS server and applies the macro associated with this event trigger.

```
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 $TRIGGER
     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
```

Table 4-1 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 4-2 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.

Related Commands

Command	Description
macro auto device	Configures macro default parameter values.
macro auto global processing	Enables Auto Smartports on a switch.
macro auto mac-address-group	Configures MAC address groups.
macro auto sticky	Configures macro persistence.
shell trigger	Creates event triggers.
show macro auto	Displays information about macros.
show shell	Displays information about event triggers and macros.

macro auto global processing

To enable Auto Smartports macros on the switch, use the **macro auto global processing** global configuration command. Use the **no** form of this command to disable the macros.

macro auto global processing

no macro auto global processing

Syntax Description

This command has no arguments or keywords.

Command Default

Auto Smartports is disabled.

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

Use the **macro auto global processing** global configuration command to globally enable macros on the *switch*. To disable macros on a specific *port*, use the **no macro auto processing** command in interface mode.

When using 802.1x or MAB authentication, you need to configure the RADIUS server to support the Cisco attribute-value pair **auto-smart-port=***event trigger*. If authentication fails, the macro is not applied. If the 802.1x or MAB authentication fails on the interface, the switch does not use the fallback CDP event trigger.

When CDP-identified devices advertise multiple capabilities, the switch chooses a capability first by switch and then by router.

To verify that a macro is applied to an interface, use the **show macro auto interface** privileged EXEC command.

Examples

This example shows how enable Auto Smartports on the switch and 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
```

Related Commands

Command	Description
macro auto device	Configures macro default parameter values.
macro auto execute	Configures mapping from an event trigger to a built-in macro.

Command	Description
macro auto mac-address-group	Configures MAC address groups.
macro auto sticky	Configures macro persistence.
shell trigger	Creates event triggers.
show macro auto	Displays information about macros.
show shell	Displays information about event triggers and macros.

macro auto mac-address-group

To create an event trigger for devices that do not support Cisco Discovery Protocol (CDP) or Link Layer Discover Protocol (LLDP), use the **macro auto mac-address-group** global configuration command. Use the **no** form of this command to delete the group.

macro auto mac-address-group *name* [**mac-address list** *list*] | [**oui** [**list** *list* | **range** *start-value* **size** *number*]]

no macro auto mac-address-group *name* [**mac-address list** *list*] | [**oui** [**list** *list* | **range** *start-value* **size** *number*]]

Syntax Description

name	Specifies the group name.
mac-address list *list*	(Optional) Configures a list of MAC addresses separated by a space.
oui	(Optional) Specifies an operationally unique identifier (OUI) **list** or **range**. • **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.

Command Default

No groups are defined.

Command Modes

Group configuration

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

Use the **macro auto mac-address-group** global configuration command to create an event trigger for devices that do not support CDP or LLDP. Use the MAC address group as a trigger to map to a built-in or user-defined macro by using the **macro auto execute** global configuration command. At link-up, the switch detects the device type and applies the specified macro.

The switch supports up to ten MAC address groups. Each group can have up to 32 OUI and 32 MAC configured addresses.

Examples

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 address-group mac 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)# end
Switch# show running configuration
!
!macro auto mac-address-group address_trigger
 oui list 333334
 oui list 333333
 oui list 233244
 oui list 455555
 mac-address list 000A.000B.000C
 mac-address list 0022.0033.0044
 mac-address list 2222.3333.3334
!

<output truncated>
```

Related Commands

Command	Description
macro auto device	Configures macro default parameter values.
macro auto execute	Configures mapping from an event trigger to a built-in macro.
macro auto global processing	Enables Auto Smartports on a switch.
macro auto sticky	Configures macro persistence.
shell trigger	Creates event triggers.
show macro auto	Displays information about macros.
show shell	Displays information about event triggers and macros.

macro auto processing

To enable Auto Smartports macros on an interface, use the **macro auto processing** interface configuration command. Use the **no** form of this command to disable the macros.

macro auto processing

no macro auto processing

Syntax Description

This command has no arguments or keywords.

Command Default

Auto Smartports is disabled.

Command Modes

Interface configuration

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

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

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 EtherChannel interface applies the configuration to the member interfaces.

To verify that a macro is applied to an interface, use the **show macro auto interface** privileged EXEC command.

Examples

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

```
Switch(config)# interface gigabitethernet 0/1
Switch(config-if)# no macro auto processing
Switch(config-if)# exit
Switch(config)# macro auto global processing
```

Related Commands

Command	Description
macro auto device	Configures macro default parameter values.
macro auto execute	Configures mapping from an event trigger to a built-in macro.
macro auto mac-address-group	Configures MAC address groups.

Command	Description
macro auto sticky	Configures macro persistence.
shell trigger	Creates event triggers.
show macro auto	Displays information about macros.
show shell	Displays information about event triggers and macros.

macro auto sticky

To configure macros to remain active after a link-down event, referred to as *macro persistence*, use the **macro auto sticky** global configuration command. Use the **no** form of this command to disable the macro persistence.

macro auto sticky

no macro auto sticky

Syntax Description

This command has no arguments or keywords.

Command Default

Macro persistence is disabled.

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

Use the **macro auto sticky** global configuration command so that macros remain active after a link-down event.

Examples

This example shows how to enable macro persistence in the global configuration mode:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config-if)# macro auto sticky
```

Related Commands

Command	Description
macro auto device	Configures macro default parameter values.
macro auto execute	Configures mapping from an event trigger to a built-in macro.

Command	Description
macro auto global processing	Enables Auto Smartports on a switch.
macro auto mac-address-group	Configures MAC address groups.
shell trigger	Creates event triggers.
show macro auto	Displays information about macros.
show shell	Displays information about event triggers and macros.

macro description

To enter a description about which macros are applied to an interface, use the **macro description** interface configuration command. Use the **no** form of this command to remove the description. This command is mandatory for Auto Smartports to work.

macro description *text*

no macro description *text*

Syntax Description

description *text*	Enters a description about the macros that are applied to the specified interface.

Command Default

None.

Command Modes

Interface configuration

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

Use the **description** keyword to associate comment text or the macro name with an interface. When multiple macros are applied on a single interface, the description text is from the last applied macro.

You can verify your settings by entering the **show parser macro description** privileged EXEC command.

Examples

This example shows how to add a description to an interface:

```
Switch(config-if)# macro description duplex settings
```

Related Commands

Command	Description
macro apply	Applies a macro on an interface.
macro global	Applies a macro on a switch or applies and traces a macro on a switch.
macro global description	Adds a description about the macros that are applied to the switch.
macro trace	Applies and traces a macro on an interface.
show parser macro	Displays the macro definition for all macros or for the specified macro.

macro global

To apply a macro to a switch or to apply and debug a macro on a switch, use the **macro global** global configuration command.

macro global {**apply** | **trace**} *macro-name* [**parameter** {*value*}] [**parameter** {*value*}] [**parameter** {*value*}]

Syntax Description

apply	Applies a macro to the switch.
trace	Applies a macro to a switch and debugs the macro.
macro-name	Specifies the name of the macro.
parameter *value*	(Optional) Specifies unique parameter values that are specific to the switch. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value.

Command Default

None.

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

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

Use the **macro global apply** *macro-name* global configuration command to apply the macro to an interface.

Use the **macro global trace** *macro-name* global configuration command to apply and then debug the macro to find any syntax or configuration errors.

If a command fails when you apply a macro because of a syntax error or a configuration error, the macro continues to apply the remaining commands to the switch.

When creating a macro that requires the assignment of unique values, use the **parameter** *value* keywords to designate values specific to the switch.

Keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value. Any full match of a keyword, even if it is part of a larger string, is considered a match and is replaced by the corresponding value.

Some macros might contain keywords that require a parameter value. You can use the **macro global apply** *macro-name* **?** command to display a list of any required values in the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.

There are Cisco-default Smartports macros embedded in the switch software. You can display these macros and the commands they contain by using the **show parser macro** user EXEC command.

Follow these guidelines when you apply a Cisco-default Smartports macro on a switch:

- Display all macros on the switch by using the **show parser macro** user EXEC command. Display the contents of a specific macro by using the **show parser macro name** *macro-name* user EXEC command.
- Keywords that begin with **$** indicate that a unique parameter value is required. Append the Cisco-default macro with the required values by using the **parameter** *value* keywords.

  The Cisco-default macros use the **$** character to help identify required keywords. There is no restriction on using the **$** character to define keywords when you create a macro.

When you apply a macro to a switch, the macro name is automatically added to the switch. You can display the applied commands and macro names by using the **show running-config** user EXEC command.

Examples

After you have created a new macro by using the **macro auto execute** global configuration command, you can apply it to a switch. This example shows how see the **snmp** macro, how to apply the macro, set the hostname to test-server, and set the IP precedence value to 7:

```
Switch# show parser macro name snmp
Macro name : snmp
Macro type : customizable

#enable port security, linkup, and linkdown traps
snmp-server enable traps port-security
snmp-server enable traps linkup
snmp-server enable traps linkdown
#set snmp-server host
snmp-server host ADDRESS
#set SNMP trap notifications precedence
snmp-server ip precedence VALUE

--------------------------------------------------
Switch(config)# macro global apply snmp ADDRESS test-server VALUE 7
```

To debug a macro, use the **macro global trace** global configuration command to find any syntax or configuration errors in the macro when you apply it to a switch. In this example, the `ADDRESS` parameter value was not entered, the `snmp-server host` command failed, and the remainder of the macro is applied to the switch:

```
Switch(config)# macro global trace snmp VALUE 7
Applying command...‘snmp-server enable traps port-security’
Applying command...‘snmp-server enable traps linkup’
Applying command...‘snmp-server enable traps linkdown’
Applying command...‘snmp-server host’
%Error Unknown error.
Applying command...‘snmp-server ip precedence 7’
```

Related Commands

Command	Description
macro apply	Applies a macro on an interface.
macro description	Adds a description about the macros that are applied to an interface.
macro global description	Adds a description about the macros that are applied to the switch.
macro trace	Applies and traces a macro on an interface.
show parser macro	Displays the macro definition for all macros or for the specified macro.

macro global description

To enter a description about the macros that are applied to a switch, use the **macro global description** global configuration command. Use the **no** form of this command to remove the description.

macro global description *text*

no macro global description *text*

Syntax Description

text	A description about the macros that are applied to the switch.

Command Default

None.

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

Use the **description** keyword to associate comment text or the macro name with a switch. When multiple macros are applied on a switch, the description text is from the last applied macro.

You can verify your settings by entering the **show parser macro description** privileged EXEC command.

Examples

This example shows how to add a description to a switch:

```
Switch(config)# macro global description udld aggressive mode enabled
```

Related Commands

Command	Description
macro apply	Applies a macro on an interface.
macro description	Adds a description about the macros that are applied to an interface.
macro global	Applies a macro on a switch or applies and traces a macro on a switch.
macro trace	Applies and debugs a macro on an interface.
show parser macro	Displays the macro definition for all macros or for the specified macro.

shell trigger

To create an event trigger, use the **shell trigger** global configuration command. Use the **no** form of this command to delete the trigger.

shell trigger *identifier description*

no shell trigger *identifier description*

Syntax Description

identifier	Specifies the event trigger identifier. The identifier should have no spaces or hyphens between words.
description	Specifies the event trigger description text.

Command Default

System-defined event triggers:

- CISCO_DMP_EVENT
- CISCO_IPVSC_AUTO_EVENT
- CISCO_PHONE_EVENT
- CISCO_SWITCH_EVENT
- CISCO_ROUTER_EVENT
- CISCO_WIRELESS_AP_EVENT
- CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

Use this command to create user-defined event triggers for use with the **macro auto device** and the **macro auto execute** global configuration commands.

To support dynamic device discovery when using IEEE 802.1x authentication, you need to configure the RADIUS authentication server to support the Cisco attribute-value pair: **auto-smart-port=***event trigger*.

Examples

This example shows how to create a user-defined event trigger called RADIUS_MAB_EVENT:

```
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)# end
```

Related Commands

Command	Description
macro auto device	Configures macro default parameter values.
macro auto execute	Configures mapping from an event trigger to a built-in macro.
macro auto global processing	Enables Auto Smartports on a switch.
macro auto mac-address-group	Configures MAC address groups.
macro auto sticky	Configures macro persistence.
shell trigger	Creates event triggers.
show macro auto	Displays information about macros.
show shell	Displays information about event triggers and macros.

show macro auto

To display Auto Smartports macro information, use the **show macro auto** user EXEC command.

show macro auto {**address-group** [*address-group-name*] | **device** [**access-point**] [**ip-camera**] [**lightweight-ap**] [**media-player**] [**phone**] [**router**] [**switch**] | **global** [*event_trigger*] | **interface** [*interface_id*]}

Syntax Description

address-group [*address-group-name*]	Displays address-group information. (Optional) *address-group-name*—Displays information for the specified address group.
device [**access-point**] [**ip-camera**] [**lightweight-ap**] [**media-player**] [**phone**] [**router**] [**switch**]	Displays device information about one or more devices. • (Optional) **access-point**—Autonomous access point • (Optional) **ip-camera**—Cisco IP video surveillance camera • (Optional) **lightweight-ap**—Lightweight access point • (Optional) **media-player**—Digital media player • (Optional) **phone**—Cisco IP phone • (Optional) **router**—Cisco router • (Optional) **switch**—Cisco switch
global [*event_trigger*]	Displays Auto Smartports information about the switch. (Optional) *event_trigger*—Displays information about the specified event trigger.
interface [*interface_id*]	Displays interface status. (Optional) *interface_id*—Displays information about the specified interface.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Usage Guidelines

Use this command to display the Auto Smartports information for the switch. Use the **show macro auto device** privileged EXEC command to display the configurable parameters for a device.

Examples

This example shows how to use the **show macro auto device** privileged EXEC command to view the configuration on the switch:

```
Switch# show macro auto device
Device:lightweight-ap
```

```
Default Macro:CISCO_LWAP_AUTO_SMARTPORT
Current Macro:CISCO_LWAP_AUTO_SMARTPORT
Configurable Parameters:ACCESS_VLAN
Defaults Parameters:ACCESS_VLAN=1
Current Parameters:ACCESS_VLAN=1

Device:access-point
Default Macro:CISCO_AP_AUTO_SMARTPORT
Current Macro:CISCO_AP_AUTO_SMARTPORT
Configurable Parameters:NATIVE_VLAN
Defaults Parameters:NATIVE_VLAN=1
Current Parameters:NATIVE_VLAN=1

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

Device:router
Default Macro:CISCO_ROUTER_AUTO_SMARTPORT
Current Macro:CISCO_ROUTER_AUTO_SMARTPORT
Configurable Parameters:NATIVE_VLAN
Defaults Parameters:NATIVE_VLAN=1
Current Parameters:NATIVE_VLAN=1

Device:switch
Default Macro:CISCO_SWITCH_AUTO_SMARTPORT
Current Macro:CISCO_SWITCH_AUTO_SMARTPORT
Configurable Parameters:NATIVE_VLAN
Defaults Parameters:NATIVE_VLAN=1
Current Parameters:NATIVE_VLAN=1

Device:ip-camera
Default Macro:CISCO_IP_CAMERA_AUTO_SMARTPORT
Current Macro:CISCO_IP_CAMERA_AUTO_SMARTPORT
Configurable Parameters:ACCESS_VLAN
Defaults Parameters:ACCESS_VLAN=1
Current Parameters:ACCESS_VLAN=1

Device:media-player
Default Macro:CISCO_DMP_AUTO_SMARTPORT
Current Macro:CISCO_DMP_AUTO_SMARTPORT
Configurable Parameters:ACCESS_VLAN
Defaults Parameters:ACCESS_VLAN=1
Current Parameters:ACCESS_VLAN=1
```

This example shows how to use the **show macro auto address-group** *name* privileged EXEC command to view the TEST3 address group configuration on the switch:

```
Switch# show macro auto address-group TEST3
MAC Address Group Configuration:

Group Name                      OUI          MAC ADDRESS
--------------------------------------------------------------
TEST3                         2233.33     0022.0022.0022
                              2233.34
```

Related Commands

Command	Description
macro auto device	Configures macro default parameter values.
macro auto execute	Configures mapping from an event trigger to a built-in macro.
macro auto global processing	Enables Auto Smartports on a switch.
macro auto mac-address-group	Configures MAC address groups.
macro auto sticky	Configures macro persistence.
shell trigger	Creates event triggers.
show shell	Displays information about event triggers and macros.

show parser macro

To display the parameters for all configured macros or for one macro on the switch, use the **show parser macro** user EXEC command.

show parser macro [{**brief** | **description** [**interface** *interface-id*] | **name** *macro-name*}]

Syntax Description

brief	(Optional) Displays the name of each macro.
description [**interface** *interface-id*]	(Optional) Displays all macro descriptions or the description of a specific interface.
name *macro-name*	(Optional) Displays information about a single macro identified by the macro name.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Examples

This is a partial output example from the **show parser macro** command. The output for the Cisco-default macros varies depending on the switch platform and the software image running on the switch:

```
Switch# show parser macro
Total number of macros = 6
--------------------------------------------------------------
Macro name : cisco-global
Macro type : default global
# Enable dynamic port error recovery for link state
# failures
errdisable recovery cause link-flap
errdisable recovery interval 60

<output truncated>

--------------------------------------------------------------
Macro name : cisco-desktop
Macro type : default interface
# macro keywords $AVID
# Basic interface - Enable data VLAN only
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID
switchport mode access

<output truncated>

--------------------------------------------------------------
Macro name : cisco-phone
Macro type : default interface
# Cisco IP phone + desktop template
# macro keywords $AVID $VVID
# VoIP enabled interface - Enable data VLAN
# and voice VLAN (VVID)
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID
switchport mode access

<output truncated>

--------------------------------------------------------------
Macro name : cisco-switch
Macro type : default interface
# macro keywords $NVID
# Access Uplink to Distribution
# Do not apply to EtherChannel/Port Group
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID

<output truncated>

--------------------------------------------------------------
Macro name : cisco-router
Macro type : default interface
# macro keywords $NVID
# Access Uplink to Distribution
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID

<output truncated>

--------------------------------------------------------------
Macro name : snmp
Macro type : customizable

#enable port security, linkup, and linkdown traps
snmp-server enable traps port-security
snmp-server enable traps linkup
snmp-server enable traps linkdown
#set snmp-server host
snmp-server host ADDRESS
#set SNMP trap notifications precedence
snmp-server ip precedence VALUE

--------------------------------------------------------------
```

This is an example of output from the **show parser macro name** command:

```
Switch# show parser macro name standard-switch10
Macro name : standard-switch10
Macro type : customizable
macro description standard-switch10
# Trust QoS settings on VOIP packets
auto qos voip trust
# Allow port channels to be automatically formed
channel-protocol pagp
```

This is an example of output from the **show parser macro brief** command:

```
Switch# show parser macro brief
    default global   : cisco-global
    default interface: cisco-desktop
    default interface: cisco-phone
    default interface: cisco-switch
    default interface: cisco-router
    customizable     : snmp
```

This is an example of output from the **show parser macro description** command:

```
Switch# show parser macro description
Global Macro(s): cisco-global
Interface    Macro Description(s)
--------------------------------------------------------------
Gi1/0/1        standard-switch10
Gi1/0/2       this is test macro
--------------------------------------------------------------
```

This is an example of output from the **show parser macro description interface** command:

```
Switch# show parser macro description interface gigabitethernet1/0/2
Interface    Macro Description
--------------------------------------------------------------
Gi1/0/2      this is test macro
--------------------------------------------------------------
```

Related Commands

Command	Description
macro apply	Applies a macro on an interface or applies and traces a macro on an interface.
macro description	Adds a description about the macros that are applied to an interface.
macro global	Applies a macro on a switch or applies and traces a macro on a switch.
macro global description	Adds a description about the macros that are applied to the switch.
show running-config	Displays the operating configuration.

show shell

Usage Guidelines

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

To display shell information, use the **show shell** user EXEC command.

show shell [**environment** | **functions** [**brief** | *shell_function*] | **triggers**]

Syntax Description

environment	(Optional) Displays shell environment information.
functions [**brief** | *shell_function*]	(Optional) Displays macro information. • **brief**—Names of the shell functions. • *shell_function*—Name of a shell function.
triggers	(Optional) Displays event trigger information.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
Cisco IOS XE 3.3SE	This command was introduced on Catalyst 3650 switches.

Examples

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_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 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=(),
                               $EC_MODE=(),$EC_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 == 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>
```

Related Commands

Command	Description
macro auto device	Configures macro default parameter values.
macro auto execute	Configures mapping from an event trigger to a built-in macro.
macro auto global processing	Enables Auto Smartports on a switch.
macro auto mac-address-group	Configures MAC address groups.
macro auto sticky	Configures macro persistence.
shell trigger	Creates event triggers.
show macro auto	Displays information about macros.