



Configuring SmartPort Macros

This chapter describes how to configure and apply SmartPort macros on your Catalyst 2950 or Catalyst 2955 switch.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the command reference for this release.

This chapter consists of these sections:

- [Understanding SmartPort Macros, page 12-1](#)
- [Configuring Smart-Port Macros, page 12-2](#)
- [Displaying SmartPort Macros, page 12-4](#)

Understanding SmartPort Macros

SmartPort macros provide a convenient way to save and share common configurations. You can use SmartPort macros to enable features and settings based on the location of a switch in the network and for mass configuration deployments across the network.

Each SmartPort macro is a set of CLI commands that you define. SmartPort macros do not contain new CLI commands; they are simply a group of existing CLI commands.

When you apply a SmartPort macro on an interface, the CLI commands within the macro are configured on the interface. When the macro is applied to an interface, the existing interface configurations are not lost. The new commands are added to interface and are saved in the running configuration file.

Cisco provides a collection of pretested, Cisco-recommended baseline configuration templates for Catalyst switches. The online reference guide templates provide the CLI commands that you can use to create SmartPort macros based on the usage of the port. You then can leverage configuration best-practices to run a converged voice, video, and data network considering quality of service (QoS), bandwidth, latency, and high performance. You can use the configuration templates to create SmartPort macros to build and deploy Cisco-recommended network designs and configurations. For more information about Cisco-recommended configuration templates, refer to the release notes for this release.

Configuring Smart-Port Macros

You can create a new SmartPort macro or use an existing macro as a template to create a new macro that is specific to your application. After you create the macro, you can apply it to an interface or a range of interfaces.

This section includes information about:

- [Default SmartPort Macro Configuration, page 12-2](#)
- [SmartPort Macro Configuration Guidelines, page 12-2](#)
- [Creating and Applying SmartPort Macros, page 12-3](#)

Default SmartPort Macro Configuration

There are no default SmartPort macros configured on the switch.

SmartPort Macro Configuration Guidelines

Follow these guidelines when configuring macros on your switch:

- Do not use **exit** or **end** commands when creating a macro. This could cause commands that follow **exit** or **end** to execute in a different command mode.
- When creating a macro, all CLI commands should be interface configuration mode commands.
- Some CLI commands are specific to certain interface types. The macro will fail the syntax check or the configuration check, and the switch will return an error message if it is applied to an interface that does not accept the configuration.
- When a macro is applied to an interface, all existing configuration on the interface is retained. This is helpful when applying an incremental configuration to an interface.
- If you modify a macro definition by adding or deleting commands, the changes are not reflected on the interface where the original macro was applied. You need to reapply the updated macro on the interface to apply the new or changed commands.
- You can use the **macro trace macro-name** interface configuration command to show what macros are running on an interface or to debug the macro to determine any syntax or configuration errors.
- If a command fails when you apply a macro, either due to a syntax error or a configuration error, the macro continues to apply the remaining commands to the interface.
- 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 individual interface within the range. If a macro command fails on one interface, it is still applied to the remaining interfaces.

Creating and Applying SmartPort Macros

Beginning in privileged EXEC mode, follow these steps to create and apply a SmartPort macro:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	macro name <i>macro-name</i>	Create a macro definition, and enter a macro name. A macro definition can contain up to 3000 characters. Enter the macro commands with one command per line. Use the @ character to end the macro. Use the # character at the beginning of a line to enter comment text within the macro. We recommend that you do not use the exit or end commands in a macro. This could cause any commands following exit or end to execute in a different command mode. For best results, all commands in a macro should be interface configuration mode commands.
Step 3	interface <i>interface-id</i>	Enter interface configuration mode, and specify the interface on which to apply the macro.
Step 4	macro {apply trace} <i>macro-name</i>	Apply each individual command defined in the macro to the interface by entering macro apply macro-name . Specify macro trace macro-name to apply and print each command before it is applied to the interface.
Step 5	macro description <i>text</i>	(Optional) Enter a description about the macro that is applied to the interface.
Step 6	end	Return to privileged EXEC mode.
Step 7	show parser macro	Verify that the macro was created.
Step 8	show running-config interface <i>interface-id</i>	Verify that the macro is applied to an interface.
Step 9	copy running-config startup-config	(Optional) Save your entries in the configuration file.

The **no** form of the **macro name** global configuration command only deletes the macro definition. It does not affect the configuration of those interfaces on which the macro is already applied. You can delete a macro-applied configuration on an interface by entering the **default interface interface-id** interface configuration command. Alternatively, you can create an *anti-macro* for an existing macro that contains the **no** form of all the corresponding commands in the original macro. Then apply the anti-macro to the interface.

This example shows how to define the **desktop-config** macro for an access switch interface, apply the macro to Fast Ethernet interface 0/9, add a description to the interface, and verify the configuration.

```
Switch(config)# macro name desktop-config
# Put the switch in access mode
switchport mode access
# Allow port to move to forwarding state quickly
spanning-tree portfast
# BPDUs should not be sent into the network
spanning-tree bpduguard enable
# Restrict the port to one address -- that of desktop
switchport port-security maximum 1
# Put all data traffic in vlan 1
switchport access vlan 1
@
```

```
Switch(config)# interface fastethernet0/9
Switch(config-if)# macro apply desktop-config
Switch(config-if)# macro description desktop-config
Switch(config-if)# end
Switch# show parser macro name desktop-config
Macro name : desktop-config
Macro type : customizable

macro description desktop-config
# Put the switch in access mode
switchport mode access
# Allow port to move to forwarding state quickly
spanning-tree portfast
# BPDUs should not be sent into the network
spanning-tree bpduguard enable
# Restrict the port to one address -- that of desktop
switchport port-security maximum 1
# Put all data traffic in vlan 1
switchport access vlan 1

Switch# show parser macro description
Interface      Macro Description
-----
Fa0/9          desktop-config
-----
```

Displaying SmartPort Macros

To display the SmartPort macros, use one or more of the privileged EXEC commands in [Table 12-1](#).

Table 12-1 Commands for Displaying SmartPort Macros

Command	Purpose
show parser macro	Displays all configured macros.
show parser macro name <i>macro-name</i>	Displays a specific macro.
show parser macro brief	Displays the configured macro names.
show parser macro description [interface <i>interface-id</i>]	Displays the macro description for all interfaces or for a specified interface.