

Configuring SmartPort Macros

**Note**

LAN Base image does *not* support SmartPorts.

This chapter describes how to configure and apply SmartPort macros on your switch. It consists of these sections:

- [Understanding SmartPort Macros, page 16-1](#)
- [Configuring SmartPort Macros, page 16-2](#)
- [Displaying SmartPort Macros, page 16-13](#)

**Note**

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

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

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

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

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 macro sets do not contain new CLI commands; Each SmartPort macro is a group of existing CLI commands.

When you apply a SmartPort macro on an interface, the CLI commands contained 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.

There are Cisco-default SmartPorts macros embedded in the switch software (see [Table 16-1](#)). You can display these macros and the commands they contain by using the **show parser macro** user EXEC command.

Table 16-1 Cisco-Default SmartPorts Macros

| Macro Name ¹ | Description |
|-------------------------|---|
| cisco-global | Use this global configuration macro to enable rapid 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 features, but with the addition of 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 using GigaStack modules or GBICs. |
| cisco-router | Use this interface configuration macro when connecting the switch and a WAN router. |

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

Cisco also 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 SmartPorts macros based on the usage of the port. You can use the configuration templates to create SmartPorts macros to build and deploy Cisco-recommended network designs and configurations.

Configuring SmartPort 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 these topics:

- [Passing Parameters Through the Macro, page 16-2](#)
- [Default SmartPort Macro Configuration, page 16-3](#)
- [SmartPort Macro Configuration Guidelines, page 16-6](#)
- [Creating SmartPorts Macros, page 16-7](#)
- [Applying SmartPorts Macros, page 16-8](#)

Passing Parameters Through the Macro

Some commands might not be sufficiently generic for all the interfaces; for example, VLAN ID for Layer 2 interfaces and the IP address for Layer 3 interface. Retaining such commands in macro definitions requires that you change the value of such parameters (like VLAN ID or IP address) before applying the macro to different interfaces. Alternatively, it requires that you create different macros for each possible value of its parameters.

The macro infrastructure can be enhanced to support accepting parameters while applying a macro. The parameters are passed as *keyword-value* pairs.

The CLI limits the number of keyword-value pairs to a maximum of three, where the first parameter must be the keyword, the second is its corresponding value, and the third parameter would be the keyword for the second keyword-value pair. Here is an example of how to pass parameters to a command-macro:

```
Switch(config)# macro name parameter-test  
Enter macro commands one per line. End with the character '@'.  
switchport mode access  
switchport access vlan $VLANID  
switchport port-security  
switchport port-security maximum $MAXHOST
```

If the above macro is applied to some interface without parameters, the invalid commands fail. Instead, you should apply the macro with appropriate keyword-value pair parameters, as follows:

```
Switch(config-if)# macro apply parameter-test $VLANID 1 $MAXHOST 5
```

The above command applies the macro after replacing \$VLANID with 1 and \$MAXHOST with 5. Be aware that you can specify any string in the macro as a keyword.

Macro Parameter Help

It is often difficult to remember the macro keywords while applying a macro to an interface or switch. Macros can contain the definitions for mandatory keywords. If you apply a macro without those keyword values, the commands are considered invalid and they will fail.

You can enhance the macro infrastructure to provide help on keywords defined in macros. While creating a macro, you can specify a help string (as a comment) to list the mandatory keywords for that macro.

The following example illustrates how to specify the help string for the keywords:

```
Switch(config)# macro name test  
switchport access vlan $VLANID  
switchport port-security maximum $MAX  
#macro keywords $VLANID $MAX
```

Help string can be anywhere in the macro. The following example illustrates an alternate way to specify the help string:

```
Switch(config)# macro name test  
switchport access vlan $VLANID  
#macro keywords $VLANID  
switchport port-security maximum $MAX  
#macro keywords $MAX
```

Default SmartPort Macro Configuration

This section illustrates the default configurations for the four supported macros. These macros can only be viewed and applied; they cannot be modified by the user.

- [cisco-global, page 16-4](#)
- [cisco-desktop, page 16-4](#)
- [cisco-phone, page 16-4](#)
- [cisco-router, page 16-5](#)
- [cisco-switch, page 16-5](#)

cisco-global

This is the example for the cisco-global macro:

```
# Enable dynamic port error recovery for link state failures.
errdisable recovery cause link-flap
errdisable recovery interval 60

# VTP requires Transparent mode for future 802.1x Guest VLAN
# and current Best Practice
vtp domain [smartports]
vtp mode transparent

# Enable aggressive mode UDLD on all fiber uplinks
udld aggressive

# Enable Rapid PVST+ and Loopguard
spanning-tree mode rapid-pvst
spanning-tree loopguard default
spanning-tree extend system-id
```

cisco-desktop

This is the example for the cisco-desktop macro:

```
# 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
# Ensure port-security age is greater than one minute
# and use inactivity timer
# "Port-security maximum 1" is the default and will not
# Show up in the config
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
```

cisco-phone

This is the example for the cisco-phone macro:

```
# 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
# Update the Voice VLAN (VVID) value which should be
# different from data VLAN
# Recommended value for voice vlan (VVID) should not be 1
switchport voice vlan $VVID
# Enable port security limiting port to a 2 MAC
# addressess -- One for desktop and two for phone
switchport port-security
switchport port-security maximum 2
```

```
# 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
# Enable auto-qos to extend trust to attached Cisco phone
auto qos voip cisco-phone
# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable@
```

cisco-router

This is the example for the cisco-router macro:

```
# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE
# Hardcode trunk and disable negotiation to
# speed up convergence
# Hardcode speed and duplex to router
switchport mode trunk
switchport nonegotiate
speed 100
duplex full
# Configure qos to trust this interface
auto qos voip trust
qos trust dscp
# Ensure fast access to the network when enabling the interface.
# Ensure that switch devices cannot become active on the interface.
spanning-tree portfast
spanning-tree bpduguard enable
```

cisco-switch

This is the example for the cisco-switch macro:

```
# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE
# Hardcode trunk and disable negotiation to
# speed up convergence
switchport mode trunk
switchport nonegotiate
# Configure qos to trust this interface
auto qos voip trust
# 802.1w defines the link as pt-pt for rapid convergence
spanning-tree link-type point-to-point
```

SmartPort Macro Configuration Guidelines

Follow these guidelines when configuring macros on your switch:

- If a command fails when you apply a macro, either due to a syntax error or to a configuration error, the macro continues to apply the remaining commands to the interface.
- **cisco-global** needs to be applied at the global configuration mode. Cisco recommends that you apply this macro before any other interface level macro.
- Specific keywords are required when you apply the system-defined macros (**cisco-desktop**, **cisco-phone**, **cisco-switch**, and **cisco-router**) on an interface.
- When using the **cisco-phone** macro to apply port security, the port security maximum is 2 (**switchport port-security maximum 2**).
- At most, 3 keyword-value pairs are allowed per system-defined macro.
- When creating a macro, do not use the **exit** or **end** commands or change the command mode by using **interface interface-id**. This could cause commands that follow **exit**, **end**, or **interface interface-id** to execute in a different command mode.
- When creating a macro, ensure that all CLI commands are in the same configuration mode.
- 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.
- Macro names are case sensitive. For example, the commands **macro name Sample-Macro** and **macro name sample-macro** result in two separate macros.
- Some macros might contain keywords that require a parameter value. You can use the **macro global apply macro-name ?** global configuration command or the **macro apply macro-name ?** interface configuration 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.
- When a macro is applied globally to a switch or to a switch interface, all existing configuration on the interface is retained. This is helpful when applying an incremental configuration.
- 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 global trace macro-name** global configuration command or the **macro trace macro-name** interface configuration command to apply and debug a macro to find any syntax or configuration errors. If a command fails because of a syntax error or a configuration error, the macro continues to apply the remaining commands.
- Some CLI commands are specific to certain interface types. If a macro is applied to an interface that does not accept the configuration, the macro will fail the syntax check or the configuration check, and the switch will return 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 macro description of the switch or interface. You can display the applied commands and macro names by using the **show parser macro description** user EXEC command.
- The user-configurable macro has a buffer that can take commands and comments up to 3000 characters. Each newline would take 2 characters and empty lines are counted as is.

There are Cisco-default SmartPorts macros embedded in the switch software (see [Table 16-1](#)). 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 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 *macro-name*** user EXEC command.
- Keywords that begin with \$ mean 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.

Creating SmartPorts Macros

To create a SmartPorts macro, follow these steps:

| | Command | Purpose |
|--------|---|--|
| Step 1 | configure terminal | Enters global configuration mode. |
| Step 2 | macro name <i>macro-name</i> | <p>Creates a macro definition, and enter a macro name. A macro definition can contain up to 3000 characters.</p> <p>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.</p> <p>Macro names are case sensitive. For example, the commands macro name Sample-Macro and macro name sample-macro result in two separate macros.</p> <p>We recommend that you do not use the exit or end commands or change the command mode by using interface <i>interface-id</i> in a macro. This could cause any commands following exit, end, or interface <i>interface-id</i> to execute in a different command mode. For best results, all commands in a macro should be in the same configuration mode.</p> <p>Note 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.</p> |
| Step 3 | end | Returns to privileged EXEC mode. |
| Step 4 | show parser macro name <i>macro-name</i> | Verifies that the macro was created. |

Applying SmartPorts Macros

To apply a SmartPorts macro, follow these steps:

| | Command | Purpose |
|--------|---|---|
| Step 1 | configure terminal | Enters global configuration mode. |
| Step 2 | macro global { apply trace } <i>macro-name</i> [parameter { <i>value</i> }] [parameter { <i>value</i> }] [parameter { <i>value</i> }] | <p>Applies each individual command defined in the macro to the switch by entering macro global apply <i>macro-name</i>. Specify macro global trace <i>macro-name</i> to apply and debug a macro to find any syntax or configuration errors.</p> <p>(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. All matching occurrences of the keyword are replaced with the corresponding value.</p> <p>Some macros might contain keywords that require a parameter value. You can use the macro global apply <i>macro-name</i> ? 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.</p> |
| Step 3 | macro global description <i>text</i> | (Optional) Enters a description about the macro that is applied to the switch. |
| Step 4 | interface <i>interface-id</i> | (Optional) Enters interface configuration mode, and specify the interface on which to apply the macro. |
| Step 5 | default interface <i>interface-id</i> | (Optional) Clears all configuration from the specified interface. |
| Step 6 | macro { apply trace } <i>macro-name</i> [parameter { <i>value</i> }] [parameter { <i>value</i> }] [parameter { <i>value</i> }] | <p>Applies each individual command defined in the macro to the interface by entering macro apply <i>macro-name</i>. Specify macro trace <i>macro-name</i> to apply and debug a macro to find any syntax or configuration errors.</p> <p>(Optional) Specify 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.</p> <p>Some macros might contain keywords that require a parameter value. You can use the macro apply <i>macro-name</i> ? 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.</p> <p>For example, here is how you apply this command:</p> <pre>Switch(config-if)# macro apply cisco-phone ? WORD Keyword to replace with a value e.g. \$AVID, \$VVID <cr></pre> |
| Step 7 | macro description <i>text</i> | (Optional) Enters a description about the macro that is applied to the interface. |
| Step 8 | end | Returns to privileged EXEC mode. |

| | Command | Purpose |
|---------|--|--|
| Step 9 | show parser macro description [interface <i>interface-id</i>] | Verifies that the macro is applied to the interface. |
| Step 10 | copy running-config startup-config | (Optional) Saves your entries in the configuration file. |

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

The **no** form of the **macro name** global configuration command deletes only 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.

The following sections illustrate how to apply and display the attachments on each of the supported macros:

- [cisco-global, page 16-9](#)
- [cisco-desktop, page 16-10](#)
- [cisco-phone, page 16-10](#)
- [cisco-switch, page 16-11](#)
- [cisco-router, page 16-12](#)

cisco-global

This example shows how to use the system-defined macro **cisco-global**:

```
Switch(config)# macro global apply cisco-global
Changing VTP domain name from gsg-switch to [smartports]
Setting device to VTP TRANSPARENT mode.
Switch(config)# end
Switch# show parser macro name cisco-global
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

# VTP requires Transparent mode for future 802.1x Guest VLAN
# and current Best Practice vtp domain [smartports] vtp mode transparent

# Enable aggressive mode UDLD on all fiber uplinks udld aggressive

# Enable Rapid PVST+ and Loopguard
spanning-tree mode rapid-pvst
spanning-tree loopguard default
spanning-tree extend system-id
```

cisco-desktop

This example shows how to use the system-defined macro **cisco-desktop** to assign a value of 35 to the access VLAN of the Fast Ethernet interface 2/9.



Note

This macro requires the **\$AVID** keyword, which is the access VLAN of the port.

```
Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-desktop $AVID 35
Switch(config-if)# end
Switch# show parser macro name cisco-desktop
Macro name : cisco-desktop
Macro type : customizable

# Basic interface - Enable data VLAN only
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID [access_vlan_id]
switchport mode access
# Enable port security limiting port to a single
# MAC address -- that of desktop
switchport port-security
# Ensure port-security age is greater than one minute
# and use inactivity timer
# "Port-security maximum 1" is the default and will not
# Show up in the config
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# show parser macro description
Interface      Macro Description
-----
Fa2/9         cisco-desktop
-----
```

cisco-phone

This example shows how to use the system-defined macro **cisco-phone** to assign a value of 35 to the access VLAN and 56 to the voice VLAN on the Fast Ethernet interface 2/9.



Note

This macro requires the **\$AVID** and **\$VVID** keywords, which are the access and voice VLANs of the port.

```
Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-phone
Switch(config-if)# macro description cisco-phone $AVID 35 $VVID 56
Switch(config-if)# end
Switch# show parser macro name cisco-phone
Macro name : cisco-phone
Macro type : customizable

# VoIP enabled interface - Enable data VLAN
# and voice VLAN (VVID)
# Recommended value for access vlan (AVID) should not be 1\
switchport access vlan $AVID [access_vlan_id]
```

```

switchport mode access
# Update the Voice VLAN (VVID) value which should be
# different from data VLAN
# Recommended value for voice vlan (VVID) should not be 1
switchport voice vlan $VVID [voice_vlan_id]
# Enable port security limiting port to a 2 MAC
# addresses -- One for desktop and one for phone
switchport port-security
switchport port-security maximum 2
# 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
# Enable auto-qos to extend trust to attached Cisco phone
auto qos voip cisco-phone
# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable@

```

```
Switch# show parser macro description
```

```
Interface      Macro Description
```

```
-----
Fa2/9          cisco-phone
-----
```

cisco-switch

This example shows how to use the system-defined macro **cisco-switch** to assign a value of 38 to the native VLAN on the Fast Ethernet interface 2/9.



Note

This macro requires the \$NVID keyword, which is the native VLANs of the port.

```

Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-switch
Switch(config-if)# macro description cisco-switch $NVID 38
Switch(config-if)# end
Switch# show parser macro name cisco-switch
Macro name : cisco-switch
Macro type : customizable

# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID [native_vlan_id]
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE [vlan_range]
# Hardcode trunk and disable negotiation to
# speed up convergence
switchport mode trunk
switchport nonegotiate
# Configure qos to trust this interface
auto qos voip trust
# 802.1w defines the link as pt-pt for rapid convergence
spanning-tree link-type point-to-point

```

```
Switch# show parser macro description
Interface      Macro Description
-----
Fa2/9         cisco-switch
-----
```

cisco-router

This example shows how to use the system-defined macro **cisco-router** to assign a value of 451 to the native VLAN on the Fast Ethernet interface 2/9.



Note

This macro requires the **\$NVID** keyword, which is the native VLANs of the port.

```
Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-router
Switch(config-if)# macro description cisco-router $NVID 45I
Switch(config-if)# end
Switch# show parser macro name cisco-router
Macro name : cisco-router
Macro type : customizable

# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID [native_vlan_id]
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE [vlan_range]
# Hardcode trunk and disable negotiation to
# speed up convergence
# Hardcode speed and duplex to router
switchport mode trunk
switchport nonegotiate
speed 100
duplex full
# Configure qos to trust this interface
auto qos voip trust
qos trust dscp
# Ensure fast access to the network when enabling the interface.
# Ensure that switch devices cannot become active on the interface.
spanning-tree portfast
spanning-tree bpduguard enable

Switch# show parser macro description
Interface      Macro Description
-----
Fa2/9         cisco-router
-----
```

Displaying SmartPort Macros

To display the SmartPort macros, use one or more of the privileged EXEC commands in [Table 16-2](#).

Table 16-2 *Commands for Displaying SmartPort Macros*

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

