

Macro Commands

This chapter contains the following sections:

- macro name, on page 2
- macro, on page 5
- macro description, on page 7
- macro global, on page 9
- macro global description, on page 11
- show parser macro, on page 12

macro name

Use the **macro name** Global Configuration mode command to define a macro. There are two types of macros that can be defined:

- Global macros define a group of CLI commands that can be run at any time.

  Smartport macros are associated with Smartport types. For each Smartport macro there must be an anti macro (a macro whose name is concatenated with **no_**). The anti macro reverses the action of the macro.

If a macro with this name already exists, it overrides the previously-defined one.

Use the **no** form of this command to delete the macro definition.

Syntax

macro name *macro-name*

no macro name [*macro-name*]

Parameters

- ***macro-name***—Name of the macro. Macro names are case sensitive.

Command Mode

Global Configuration mode

User Guidelines

A macro is a script that contains CLI commands and is assigned a name by the user. It can contain up to 3000 characters and 200 lines.

Keywords

Macros may contain keywords (parameters). The following describes these keywords:

- A macro can contain up to three keywords.
- All matching occurrences of the keyword are replaced by the corresponding value specified in the **macro** command.
- Keyword matching is case-sensitive

  Applying a macro with keywords does not change the state of the original macro definition.

  User Feedback

The behavior of a macro command requiring user feedback is the same as if the command is entered from terminal: it sends its prompt to the terminal and accepts the user reply.

Creating a Macro

Use the following guidelines to create a macro:

- Use **macro name** to create the macro with the specified name.

- Enter one macro command per line.
- Use the **@** character to end the macro.
- Use the **#** character at the beginning of a line to enter a comment in the macro.

  In addition, # is used to identify certain preprocessor commands that can only be used within a macro. There are two possible preprocessor commands:

  #macro key description - Each macro can be configured with up to 3 keyword/description pairs. The keywords and descriptions are displayed in the GUI pages when the macro is displayed.

  The syntax for this preprocessor command is as follows:

  #macro key description $*keyword1 description1 $keyword2 description2 $keyword3 description3*

  A keyword must be prefixed with '$'.

  #**macro keywords -** This instruction enables the device to display the keywords as part of the CLI help. It accepts up to 3 keywords. The command creates a CLI help string with the keywords for the macro. The help string will be displayed if help on the macro is requested from the **macro** and **macro global** commands. The GUI also uses the keywords specified in the command as the parameter names for the macro. See Example 2 and 3 below for a description of how this command is used in the CLI.

  The syntax for this preprocessor command is as follows:

  #macro keywords $*keyword1* $*keyword2* $*keyword3*

  where $keywordn is the name of the keyword.

  Editing a Macro

Macros cannot be edited. Modify a macro by creating a new macro with the same name as the existing macro. The newer macro overwrites the existing macro.

The exceptions to this are the built-in macros and corresponding anti-macros for the Smartport feature. You cannot override a Smartport macro.

Scope of Macro

It is important to consider the scope of any user-defined macro. Because of the potential hazards of applying unintended configurations, do not change configuration modes within the macro by using commands such as **exit**, **end**, or **interface** *interface-id*. With a few exceptions, there are other ways of executing macros in the various configuration modes. Macros may be executed in Privileged Exec mode, Global Configuration mode, and Interface Configuration mode (when the interface is NOT a VLAN.)

Example 1 -The following example shows how to create a macro that configures the duplex mode of a port.

```
switchxxxxxx(config)# macro name dup
Enter macro commands one per line. End with the character '@'.
#macro description dup
duplex full
negotiation
@
```

Example 2 -The following example shows how to create a macro with the parameters: DUPLEX and SPEED. When the macro is run, the values of DUPLEX and SPEED must be provided by the user. The **#macro keywords** command enables the user to receive help for the macro as shown in Example 3.

```
switchxxxxxx(config)# macro name duplex
Enter macro commands one per line. End with the character '@'.
duplex $DUPLEX
no negotiation
speed $SPEED
#macro keywords $DUPLEX $SPEED
@
```

Example 3 -The following example shows how to display the keywords using the help character ? (as defined by the **#macro keywords** command above) and then run the macro on the port. The **#macro keywords** command entered in the macro definition enables the user to receive help for the macro, as shown after the words e.g. below.

```
switchxxxxxx(config)# interface gi1/0/1
switchxxxxxx(config-if)# macro apply duplex ?
    WORD <1-32>   Keyword to replace with value e.g. $DUPLEX, $SPEED
    <cr>
switchxxxxxx(config-if)# macro apply duplex $DUPLEX ?
    WORD<1-32>   First parameter value
    <cr>
switchxxxxxx(config-if)# macro apply duplex $DUPLEX full $SPEED ?
    WORD<1-32>   Second parameter value
switchxxxxxx(config-if)# macro apply duplex $DUPLEX full $SPEED 100
```

macro

Use the **macro apply/trace** Interface Configuration command to either:

- Apply a macro to an interface without displaying the actions being performed
- Apply a macro to the interface while displaying the actions being performed

Syntax

macro {**apply** | **trace**} *macro-name* [*parameter-name1 value*] [*parameter-name2 value*] [*parameter-name3 value*]

Parameters

- **apply**—Apply a macro to the specific interface.
- **trace**—Apply and trace a macro to the specific interface.
- ***macro-name***—Name of the macro.
- ***parameter-name*** value—For each parameter defined in the macro, specify its name and value. You can enter up to three parameter-value pairs. Parameter keyword matching is case sensitive. All matching occurrences of the parameter name in the macro are replaced with the corresponding value.

Default Configuration

The command has no default setting.

Command Mode

Interface (Ethernet, Port Channel) Configuration mode

User Guidelines

The **macro apply** command hides the commands of the macro from the user while it is being run. The **macro trace** command displays the commands along with any errors which are generated by them as they are executed. This is used to debug the macro and find syntax or configuration errors.

When you run a macro, if a line in it fails because of a syntax or configuration error, the macro continues to apply the remaining commands to the interface.

If you apply a macro that contains parameters in its commands, the command fails if you do not provide the values for the parameters. You can use the **macro apply** macro-name with a '**?**' to display the help string for the macro keywords (if you have defined these with the **#macro keywords** preprocessor command).

Parameter (keyword) matching is case sensitive. All matching occurrences of the parameter are replaced with the provided value. Any full match of a keyword, even if it is part of a large string, is considered a match and replaced by the corresponding value.

When you apply a macro to an interface, the switch automatically generates a macro description command with the macro name. As a result, the macro name is appended to the macro history of the interface. The **show parser macro** command displays the macro history of an interface.

A macro applied to an interface range behaves the same way as a macro applied to a single interface. When a macro is applied to an interface range, it is applied sequentially to each interface within the range. If a macro command fails on one interface, it is nonetheless attempted to be applied and may fail or succeed on the remaining interfaces.

Example 1 - The following is an example of a macro being applied to an interface with the trace option.

```
switchxxxxxx(config)# interface gi1/0/2
switchxxxxxx(config-if)# macro trace dup $DUPLEX full $SPEED 100
  Applying command…  'duplex full'
  Applying command…  'speed 100'
switchxxxxxx(config-if)#
```

Example 2 - The following is an example of a macro being applied without the trace option.

```
switchxxxxxx(config)# interface gi1/0/2
switchxxxxxx(config-if)# macro apply dup $DUPLEX full $SPEED 100
switchxxxxxx(config-if)#
```

Example 3 - The following is an example of an incorrect macro being applied.

```
switchxxxxxx(config)# interface gi1/0/1
switchxxxxxx(config-if)# macro trace dup
Applying command...'duplex full'
Applying command...'speed auto'
% bad parameter value
switchxxxxxx(config-if)#
```

macro description

Use the **macro description** Interface Configuration mode command to append a description, for example, a macro name, to the macro history of an interface. Use the **no** form of this command to clear the macro history of an interface. When the macro is applied to an interface, the switch automatically generates a macro description command with the macro name. As a result, the name of the macro is appended to the macro history of the interface.

Syntax

macro description text

no macro description

Parameters

- ***text***—Description text. The text can contain up to 160 characters. The text must be double quoted if it contains multiple words.

Default Configuration

The command has no default setting.

Command Mode

Interface (Ethernet, Port Channel) Configuration mode

User Guidelines

When multiple macros are applied on a single interface, the description text is a concatenation of texts from a number of previously-applied macros.

Example

```
switchxxxxxx(config)# interface gi1/0/2
switchxxxxxx(config-if)# macro apply dup
switchxxxxxx(config-if)# exit
switchxxxxxx(config)# interface gi1/0/3
switchxxxxxx(config-if)# macro apply duplex $DUPLEX full $SPEED 100
switchxxxxxx(config-if)# macro description dup
switchxxxxxx(config-if)# macro description duplex
switchxxxxxx(config-if)# end
switchxxxxxx(config)# exit
switchxxxxxx# show parser macro description
Global Macro(s):
Interface      Macro Description(s)
------------   ------------------------------------------------
gi1/0/2             dup
gi1/0/3             duplex | dup | duplex
-----------------------------------------------------------------
switchxxxxxx# configure
switchxxxxxx(config)# interface gi1/0/2
switchxxxxxx(config-if)# no macro description
switchxxxxxx(config-if)# end
switchxxxxxx(config)# exit
switchxxxxxx# show parser macro description
```

```
Global Macro(s):
Interface      Macro Description(s)
---------      ------------------------------------------------------
gi1/0/3            duplex | dup | duplex
---------------------------------------------------------------
```

macro global

Use the **macro global** Global Configuration command to apply a macro to a switch (with or without the trace option).

Syntax

macro global {**apply** | **trace**} *macro-name* [*parameter-name1 value*] [*parameter-name2 value*] *[parameter -name3 value*]

Parameters

- **apply**—Apply a macro to the switch.
- **trace**—Apply and trace a macro to the switch.
- ***macro-name***—Specify the name of the macro.
- ***parameter-name*** value—Specify the parameter values required for the switch. You can enter up to three parameter-value pairs. Parameter keyword matching is case sensitive. All matching occurrences of the parameters are replaced with the corresponding value.

Default Configuration

The command has no default setting.

Command Mode

Global Configuration mode.

User Guidelines

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 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 large string, is considered a match and replaced by the corresponding value.

If you apply a macro that contains keywords in its commands, the command fails if you do not specify the proper values for the keywords when you apply the macro. You can use this command with a '?' to display the help string for the macro keywords. You define the keywords in the help string using the preprocessor command **#macro keywords** when you define a macro.

When you apply a macro in Global Configuration mode, the switch automatically generates a global macro description command with the macro name. As a result, the macro name is appended to the global macro history.

Example

The following is an example of a macro being defined and then applied to the switch with the trace option.

```
switchxxxxxx(config)# macro name console-timeout
Enter macro commands one per line. End with the character '@'.
```

```
line console
exec-timeout $timeout-interval
@
switchxxxxxx(config)# macro global trace console-timeout $timeout-interval 100
  Applying command…  'line console'
  Applying command…  'exec-timeout 100'
```

macro global description

Use the **macro global description** Global Configuration command to enter a description which is used to indicate which macros have been applied to the switch. Use the **no** form of this command to remove the description.

Syntax

macro global description text

no macro global description

Parameters

- ***text***—Description text. The text can contain up to 160 characters.

Default Configuration

The command has no default setting.

Command Mode

Global Configuration mode

User Guidelines

When multiple global macros are applied to a switch, the global description text is a concatenation of texts from a number of previously applied macros.

Examples

```
switchxxxxxx(config)# macro global description "set console timeout interval"
```

show parser macro

Use the **show parser macro** User EXEC mode command to display the parameters for all configured macros or for one macro on the switch.

Syntax

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

Parameters

- **brief**—Display the name of all macros.
- **description** [**interface** interface-id]—Display the macro descriptions for all interfaces or if an interface is specified, display the macro descriptions for that interface.
- **name** macro-name—Display information about a single macro identified by the macro name.
- **detailed**—Displays information for non-present ports in addition to present ports.

Default Configuration

Display description of all macros on present ports.

If the **detailed** keyword is not used, only present ports are displayed.

Command Mode

User EXEC mode

Example 1 - This is a partial output example from the **show parser macro** command.

```
switchxxxxxx# show parser macro
Total number of macros = 6
----------------------------------------------------------------
Macro name : company-global
Macro type : default global
# Enable dynamic port error recovery for link state
# failures
----------------------------------------------------------------
Macro name : company-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
```

Example 2 - This is an example of output from the **show parser macro name** command.

```
switchxxxxxx# show parser macro 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
```

Example 3 - This is an example of output from the **show parser macro brief** command.

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

Example 4 - This is an example of output from the **show parser macro description** command.

```
switchxxxxxx# show parser macro description
Global Macro(s): company-global
```

Example 5 - This is an example of output from the **show parser macro description interface** command.

```
switchxxxxxx# show parser macro description interface gi1/0/2
Interface Macro Description
---------------------------------------------------------------
gi1/0/2 this is test macro
---------------------------------------------------------------
```