Managing Switch Stacks

Prerequisites for Switch Stacks

- All the switches in the stack must be running the same license level as the active switch. For information about license levels, see the *System Management* section of this guide.
- All the switches in the switch stack must be running compatible software versions.

Restrictions for Switch Stacks

The following are restrictions for your switch stack configuration:

- Only homogenous stacking is supported, that is, a stack of Cisco Catalyst 9300 Series Switches stack with only Cisco Catalyst 9300 Series Switches as stack members.

  Cisco Catalyst 9300L Series Switches stack with only Cisco Catalyst 9300L Series Switches as stack members.

  C9300-24UB, C9300-24UXB, and C9300-48UB switches can only be stacked with each other.

- During a switchover, when the standby device syncs with the active device, the following log message is displayed on the console:

  ```
  %SM-4-BADEVENT: Event 'standby_phy_link_up' is invalid for
  the current state 'NO_NEIGHBOR': rep_lsl_rx Gix/x/x -Traceback=
  ```

  Ignore this message. It does not have any functional or operational impact.

- You cannot have a switch stack containing a mix of different license levels.

Information About Switch Stacks

Switch Stack Overview

A switch stack can have up to eight stacking-capable switches connected through their StackWise-480 ports. Note that Cisco Catalyst 9300L series switches connect through their StackWise-320 ports.The stack members work together as a unified system. Layer 2 and Layer 3 protocols present the entire switch stack as a single entity to the network.

A switch stack always has one active switch and one standby switch. If the active switch becomes unavailable, the standby switch assumes the role of the active switch, and continues to the keep the stack operational.

The active switch controls the operation of the switch stack, and is the single point of stack-wide management.

From the active switch, you configure:

- System-level (global) features that apply to all stack members
- Interface-level features for each stack member

The active switch contains the saved and running configuration files for the switch stack. The configuration files include the system-level settings for the switch stack and the interface-level settings for each stack member. Each stack member has a current copy of these files for back-up purposes.

Switch Stack Membership

A standalone is a stack with one stack member that also operates as the active switch. You can connect one standalone to another to create a stack containing two stack members, with one of them as the active switch. You can connect standalone to an existing stack to increase the stack membership.

Hello messages are sent and received by all stack members.

- If a stack member does not respond, that member is removed from the stack.
- If the standby does not respond, a new standby is elected.
- If the active does not respond, the standby becomes the active .

In addition, keepalive messages are sent and received between the active and standby es.

- If the standby does not respond, a new standby is elected.
- If the active does not respond, the standby becomes the active .

Changes to Switch Stack Membership

If you replace a stack member with an identical model, the new switch functions with exactly the same configuration as the replaced switch, assuming that the new switch (referred to as the provisioned switch) is using the same member number as the replaced switch.

The operation of the switch stack continues uninterrupted during membership changes unless you remove the active switch or you add powered-on standalone switches or switch stacks.

- Adding powered-on switches (merging) causes all switches to reload and elect a new active switch from among themselves. The newly elected active switch retains its role and configuration. All other switches retain their stack member numbers and use the stack configuration of the newly elected active switch.
- Removing powered-on stack members causes the switch stack to divide (partition) into two or more switch stacks, each with the same configuration. This can cause:
  - An IP address conflict in your network. If you want the switch stacks to remain separate, change the IP address or addresses of the newly created switch stacks.
  - A MAC address conflict between two members in the stack. You can use the **stack-mac update force** command to resolve the conflict.

If a newly created switch stack does not have an active switch or standby switch, the switch stack will reload and elect a new active switch.

Note Make sure that you power off the switches that you add to or remove from the switch stack.

After adding or removing stack members, make sure that the switch stack is operating at full bandwidth. Press the Mode button on a stack member until the Stack mode LED is on. The last two right port LEDs on all switches in the stack should be green. Depending on the switch model, the last two right ports are 10-Gigabit Ethernet ports or small form-factor pluggable (SFP) module ports (10/100/1000 ports). If one or both of these LEDs are not green on any of the switches, the stack is not operating at full bandwidth.

If you remove powered-on members but do not want to partition the stack:

- Power off the switches in the newly created switch stacks.
- Reconnect them to the original switch stack through their stack ports.
- Power on the switches.

For cabling and power considerations that affect switch stacks, see the *Cisco Catalyst 9300 Series Switches Hardware Installation Guide* .

Stack Member Numbers

The stack member number (1 to 8) identifies each member in the switch stack. The member number also determines the interface-level configuration that a stack member uses. You can display the stack member number by using the **show switch** EXEC command.

A new, out-of-the-box switch (one that has not joined a switch stack or has not been manually assigned a stack member number) ships with a default stack member number of 1. When it joins a switch stack, its default stack member number changes to the lowest available member number in the stack.

Stack members in the same switch stack cannot have the same stack member number. Every stack member, including a standalone switch, retains its member number until you manually change the number or unless the number is already being used by another member in the stack.

- If you manually change the stack member number by using the **switch** *current-stack-member-number* **renumber** *new-stack-member-number* EXEC command, the new number goes into effect after that stack member resets (or after you use the **reload slot** *stack-member-number* privileged EXEC command) and

only if that number is not already assigned to any other members in the stack. Another way to change the stack member number is by changing the SWITCH_NUMBER environment variable.

If the number is being used by another member in the stack, the switch selects the lowest available number in the stack.

If you manually change the number of a stack member and no interface-level configuration is associated with that new member number, that stack member resets to its default configuration.

You cannot use the **switch** *current-stack-member-number* **renumber** *new-stack-member-number* EXEC command on a provisioned switch. If you do, the command is rejected.

- If you move a stack member to a different switch stack, the stack member retains its number only if the number is not being used by another member in the stack. If it is being used, the switch selects the lowest available number in the stack.
- If you merge switch stacks, the switch that join the switch stack of a new active switch select the lowest available numbers in the stack.

As described in the hardware installation guide, you can use the switch port LEDs in Stack mode to visually determine the stack member number of each stack member.

You can enter the Stack mode on any of these switches by pressing the mode button. Based on the switch number configured on each switch, the corresponding port LED will be blinking green. For instance, if the switch number configured on a particular switch is three, then the port LED-3 will be blinking green when the mode button is set to stack.

Stack Member Priority Values

A higher priority value for a stack member increases the probability of it being elected active switch and retaining its stack member number. The priority value can be 1 to 15. The default priority value is 1. You can display the stack member priority value by using the **show switch** EXEC command.

Note We recommend assigning the highest priority value to the that you prefer to be the active switch. This ensures that the is reelected as the active switch if a reelection occurs.

To change the priority value for a stack member, use the **switch** *stack-member-number* **priority** *new priority-value* EXEC command. For more information, see the “Setting the Stack Member Priority Value” section.

The new priority value takes effect immediately but does not affect the current active switch. The new priority value helps determine which stack member is elected as the new active switch when the current active switch or the switch stack resets.

Switch Stack Bridge ID and MAC Address

A switch stack is identified in the network by its *bridge ID* and, if it is operating as a Layer 3 device, its router MAC address. The bridge ID and router MAC address are determined by the MAC address of the active switch.

If the active switch changes, the MAC address of the new active switch determines the new bridge ID and router MAC address.

If the entire switch stack reloads, the switch stack uses the MAC address of the active switch.

Persistent MAC Address on the Switch Stack

You can use the persistent MAC address feature to set a time delay before the stack MAC address changes. During this time period, if the previous active switch rejoins the stack, the stack continues to use its MAC address as the stack MAC address, even if the switch is now a stack member and not an active switch. If the previous active switch does not rejoin the stack during this period, the switch stack takes the MAC address of the new active switch as the stack MAC address. By default, the stack MAC address will be the MAC address of the first active switch, even if a new active switch takes over.

Note You can also configure stack MAC persistency so that the stack MAC address never changes to the new active switch MAC address, by using the **stack-mac persistent timer 0** command.

Active and Standby Switch Election and Reelection

All stack members are eligible to be the active switch or the standby switch. If the active switch becomes unavailable, the standby switch becomes the active switch.

An active switch retains its role unless one of these events occurs:

- The switch stack is reset.
- The active switch is removed from the switch stack.
- The active switch is reset or powered off.
- The active switch fails.
- The switch stack membership is increased by adding powered-on standalone switches or switch stacks.

The active switch is elected or reelected based on one of these factors and in the order listed:

1. The switch that is currently the active switch.
2. The switch with the highest stack member priority value.

Note We recommend assigning the highest priority value to the switch that you prefer to be the active switch. This ensures that the switch is reelected as active switch if a reelection occurs.

3. The switch with the shortest start-up time.
4. The switch with the lowest MAC address.

Note The factors for electing or reelecting a new standby switch are same as those for the active switch election or reelection, and are applied to all participating switches except the active switch.

After election, the new active switch becomes available after a few seconds. In the meantime, the switch stack uses the forwarding tables in memory to minimize network disruption. The physical interfaces on the other available stack members are not affected during a new active switch election and reset.

When the previous active switch becomes available, it *does not* resume its role as the active switch.

If you power on or reset an entire switch stack, some stack members *might not* participate in the active switch election. Stack members that are powered on within the same 2-minute timeframe participate in the active switch election and have a chance to become the active switch. Stack members that are powered on after the 120-second timeframe do not participate in this initial election and become stack members. For powering considerations that affect active-switch elections, see the switch hardware installation guide.

As described in the hardware installation guide, you can use the ACTV LED on the switch to see if the switch is the active switch.

Switch Stack Configuration Files

The active switch has the saved and running configuration file for the switch stack. The standby switch automatically receives the synchronized running configuration file. Stack members receive synchronized copies when the running configuration file is saved into the startup configuration file. If the active switch becomes unavailable, the standby switch takes over with the current running configuration.

The configuration files record these settings:

- System-level (global) configuration settings such as IP, STP, VLAN, and SNMP settings that apply to all stack members
- Stack member interface-specific configuration settings that are specific for each stack member

Note The interface-specific settings of the active switch are saved if the active switch is replaced without saving the running configuration to the startup configuration.

A new, out-of-box switch joining a switch stack uses the system-level settings of that switch stack. If a switch is moved to a different switch stack before it is powered on, that switch loses its saved configuration file and uses the system-level configuration of the new switch stack. If the switch is powered on as a standalone switch before it joins the new switch stack, the stack will reload. When the stack reloads, the new switch may become the active switch, retain its configuration and overwrite the configuration files of the other stack members.

The interface-specific configuration of each stack member is associated with the stack member number. Stack members retain their numbers unless they are manually changed or they are already used by another member in the same switch stack. If the stack member number changes, the new number goes into effect after that stack member resets.

- If an interface-specific configuration does not exist for that member number, the stack member uses its default interface-specific configuration.
- If an interface-specific configuration exists for that member number, the stack member uses the interface-specific configuration associated with that member number.

If you replace a failed member with an identical model, the replacement member automatically uses the same interface-specific configuration as the failed switch. You do not need to reconfigure the interface settings. The replacement switch (referred to as the provisioned switch) must have the same stack member number as the failed switch.

You back up and restore the stack configuration in the same way as you would for a standalone switch configuration.

Offline Configuration to Provision a Stack Member

You can use the offline configuration feature to *provision* (to supply a configuration to) a new switch before it joins the switch stack. You can configure the stack member number, the switch type, and the interfaces associated with a switch that is not currently part of the stack. The configuration that you create on the switch stack is called the *provisioned configuration*. The switch that is added to the switch stack and that receives this configuration is called the *provisioned switch*.

You manually create the provisioned configuration through the **switch** *stack-member-number* **provision** *type* global configuration command. You must change the *stack-member-number* on the provisioned switch before you add it to the stack, and it must match the stack member number that you created for the new switch on the switch stack. The switch type in the provisioned configuration must match the switch type of the newly added switch. The provisioned configuration is automatically created when a switch is added to a switch stack and when no provisioned configuration exists.

When you configure the interfaces associated with a provisioned switch, the switch stack accepts the configuration, and the information appears in the running configuration. However, as the switch is not active, any configuration on the interface is not operational and the interface associated with the provisioned switch does not appear in the display of the specific feature. For example, VLAN configuration information associated with a provisioned switch does not appear in the **show vlan** user EXEC command output on the switch stack.

The switch stack retains the provisioned configuration in the running configuration whether or not the provisioned switch is part of the stack. You can save the provisioned configuration to the startup configuration file by entering the **copy running-config startup-config** privileged EXEC command. The startup configuration file ensures that the switch stack can reload and can use the saved information whether or not the provisioned switch is part of the switch stack.

Upgrading a Switch Running Incompatible Software

The auto-upgrade and auto-advise features enable a switch with software packages that are incompatible with the switch stack to be upgraded to a compatible software version so that it can join the switch stack.

Switch Stack Management Connectivity

You manage the switch stack and the stack member interfaces through the active switch. You can use the CLI, SNMP, and any of the supported network management applications. You cannot manage stack members on an individual basis.

Note Use SNMP to manage network features across the stack that are defined by supported MIBs. The switch does not support MIBs to manage stacking-specific features such as stack membership and election.

How to Configure a Switch Stack

Temporarily Disabling a Stack Port

If a stack port is flapping and causing instability in the stack ring, to disable the port, enter the **switch** *stack-member-number* **stack port** *port-number* **disable** privileged EXEC command. To reenable the port, enter the **switch** *stack-member-number* **stack port** *port-number* **enable** command.

Note Be careful when using the **switch** *stack-member-number* **stack port** *port-number* **disable** command. When you disable the stack port, the stack operates at half bandwidth.

A stack is in the full-ring state when all members are connected through the stack ports and are in the ready state.

The stack is in the partial-ring state when the following occurs:

- All members are connected through their stack ports but some are not in the ready state.
- Some members are not connected through the stack ports.

Procedure

	Command or Action	Purpose
Step 1	**switch** *stack-member-number* **stack port** *port-number* **disable** **Example:** `# switch 2 stack port 1 disable`	Disables the specified stack port.
Step 2	**switch** *stack-member-number* **stack port** *port-number* **enable** **Example:** `# switch 2 stack port 1 enable`	Reenables the stack port.

When you disable a stack port and the stack is in the full-ring state, you can disable only one stack port. This message appears:

```
Enabling/disabling a stack port may cause undesired stack changes. Continue?[confirm]
```

When you disable a stack port and the stack is in the partial-ring state, you cannot disable the port. This message appears:

```
Disabling stack port not allowed with current stack configuration.
```

Reenabling a Stack Port While Another Member Starts

Stack Port 1 on Switch 1 is connected to Port 2 on Switch 4. If Port 1 is flapping, you can disable Port 1 with the **switch 1 stack port 1 disable** privileged EXEC command. While Port 1 on Switch 1 is disabled and Switch 1 is still powered on, follow these steps to reenable a stack port:

Procedure

Step 1 Disconnect the stack cable between Port 1 on Switch 1 and Port 2 on Switch 4.

Step 2 Remove Switch 4 from the stack.

Step 3 Add a switch to replace Switch 4 and assign it switch-number 4.

Step 4 Reconnect the cable between Port 1 on Switch 1 and Port 2 on Switch 4 (the replacement switch).

Step 5 Reenable the link between the switches. Enter the **switch 1 stack port 1 enable** privileged EXEC command to enable Port 1 on Switch 1.

Step 6 Power on Switch 4.

Caution Powering on Switch 4 before enabling the Port 1 on Switch 1 might cause one of the switches to reload.

If Switch 4 is powered on first, you might need to enter the **switch 1 stack port 1 enable** and the **switch 4 stack port 2 enable** privileged EXEC commands to bring up the link.

Monitoring the Device Stack

Table 1: Commands for Displaying Stack Information

Command	Description
show switch	Displays summary information about the stack, including the status of provisioned switches and switches in version-mismatch mode.
show switch *stack-member-number*	Displays information about a specific member.
show module	Displays summary informaton about the stack.
show switch detail	Displays detailed information about the stack.
show switch neighbors	Displays the stack neighbors.
show switch stack-ports [**summary**]	Displays port information for the stack. Use the **summary** keyword to display the stack cable length, the stack link status, and the loopback status.
show switch stack-ports [**detail**]	Displays the stack link status and information for each stack member. Use the **detail** keyword to display the stack interface status, errors, drops, packet transmission and bandwidth details.

Command	Description
show redundancy	Displays the redundant system and the current processor information. The redundant system information includes the system uptime, standby failures, switchover reason, hardware, configured and operating redundancy mode. The current processor information displayed includes the active location, the software state, the uptime in the current state and so on.
show redundancy state	Displays all the redundancy states of the active and standby devices.

Configuration Examples for Switch Stacks

Switch Stack Configuration Scenarios

Most of these switch stack configuration scenarios assume that at least two devices are connected through their StackWise ports.

Table 2: Configuration Scenarios

Scenario		Result
Active switch election specifically determined by existing active switches	Connect two powered-on switch stacks through the StackWise ports.	Only one of the two active switches becomes the new active switch.
Active switch election specifically determined by the stack member priority value	**1.** Connect two switches through their StackWise ports. **2.** Use the **switch** *stack-member-number* **priority** *new-priority-number* EXEC command to set one stack member with a higher member priority value. **3.** Restart both member switches at the same time.	The stack member with the higher priority value is elected active switch.
Active switch election specifically determined by the configuration file	Assuming that both member switches have the same priority value: **1.** Make sure that one stack member has a default configuration and that the other stack member has a saved (nondefault) configuration file. **2.** Restart both member switches at the same time.	The stack member with the saved configuration file is elected active switch.

Scenario		Result
Active switch election specifically determined by the MAC address	Assuming that both member switches have the same priority value, configuration file, and license level, restart both member switches at the same time.	The stack member with the lower MAC address is elected active switch .
Stack member number conflict	Assuming that one stack member has a higher priority value than the other stack member: **1.** Ensure that both member switches have the same stack member number. If necessary, use the **switch** *current-stack-member-number* **renumber** *new-stack-member-number* EXEC command. **2.** Restart both member switches at the same time.	The stack member with the higher priority value retains its stack member number. The other stack member has a new stack member number.
Add a stack member	**1.** Power off the new switch. **2.** Through their StackWise ports, connect the new switch to a powered-on switch stack. **3.** Power on the new switch.	The active switch is retained. The new switch is added to the switch stack.
Active switch failure	Remove (or power off) the active switch.	The standby switch becomes the new active switch. All other member switches in the stack remain as member switches and do not reboot.
Add eight member switches	**1.** Through their StackWise ports, connect eight devices. **2.** Power on all devices.	Two devices become active switches. One active switch has eight member switches. The other active switch remains as a standalone device. Use the Mode button and port LEDs on the device to identify which devices are active switches and which devices belong to each active switch.

Enabling the Persistent MAC Address Feature: Example

This example shows how to configure the persistent MAC address feature for a 7-minute time delay and to verify the configuration:

```
Device(config)# stack-mac persistent timer 7
  WARNING: The stack continues to use the base MAC of the old active
  WARNING: as the stack-MAC after a active switchover until the MAC
  WARNING: persistency timer expires. During this time the Network
  WARNING: Administrators must make sure that the old stack-mac does
```

```
WARNING: not appear elsewhere in this network domain. If it does,
WARNING: user traffic may be blackholed.
Device(config)# end
Device# show switch
Switch/Stack Mac Address : 0016.4727.a900
Mac persistency wait time: 7 mins
                                           H/W   Current
Switch#  Role   Mac Address     Priority Version  State
-----------------------------------------------------------
*1       Active 0016.4727.a900     1      P2B     Ready
```

Provisioning a New Member for a Switch Stack: Example

The **show running-config** command output shows the interfaces associated with the provisioned switch:

```
(config)# switch 2 provision switch_PID
  (config)# end
  # show running-config | include switch 2
  !
  interface GigabitEthernet2/0/1
  !
  interface GigabitEthernet2/0/2
  !
  interface GigabitEthernet2/0/3
  <output truncated>
```

show switch stack-ports summary Command Output: Example

Only Port 1 on stack member 2 is disabled.

```
# show switch stack-ports summary
  #/  Stack   Neighbor   Cable    Link   Link   Sync      #         In
   Port#     Port              Length    OK   Active   OK    Changes   Loopback
            Status                                        To LinkOK
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     OK         3      50 cm     Yes    Yes    Yes        1        No
    1/2     Down      None    3 m       Yes    No     Yes        1        No
    2/1     Down      None    3 m       Yes    No     Yes        1        No
    2/2     OK         3      50 cm     Yes    Yes    Yes        1        No
    3/1     OK         2      50 cm     Yes    Yes    Yes        1        No
    3/2     OK         1      50 cm     Yes    Yes    Yes        1        No
```

Table 3: show switch stack-ports summary Command Output

Field	Description
Switch#/Port#	Member number and its stack port number.
Stack Port Status	Status of the stack port. • Down—A cable is detected, but either no connected neighbor is up, or the stack port is disabled. • OK—A cable is detected, and the connected neighbor is up.

Field	Description
Neighbor	Switch number of the active member at the other end of the stack cable.
Cable Length	Valid lengths are 50 cm, 1 m, or 3 m. If the switch cannot detect the cable length, the value is *no cable*. The cable might not be connected, or the link might be unreliable. When there is no cable connected to the stack port, the value displayed is *no cable* along with the cable length value.
Link OK	Whether the stack cable is connected and functional. There may or may not be a neighbor connected on the other end. The *link partner* is a stack port on a neighbor switch. • No—There is no stack cable connected to this port or the stack cable is not functional. • Yes—There is a functional stack cable connected to this port.
Link Active	Whether a neighbor is connected on the other end of the stack cable. • No—No neighbor is detected on the other end. The port cannot send traffic over this link. • Yes—A neighbor is detected on the other end. The port can send traffic over this link.
Sync OK	Whether the link partner sends valid protocol messages to the stack port. • No—The link partner does not send valid protocol messages to the stack port. • Yes—The link partner sends valid protocol messages to the port.
# Changes to LinkOK	The relative stability of the link. If a large number of changes occur in a short period of time, link flapping can occur.
In Loopback	Whether a stack cable is attached to a stack port on the member. • No—At least one stack port on the member has an attached stack cable. • Yes—None of the stack ports on the member has an attached stack cable.

show switch stack-ports detail Command Output: Example

The following is a sample output of the command for a working stack:

```
Device# show switch stack-ports detail
1/1 is DOWN Loopback No
Cable Length 50cm     Neighbor NONE
Link Ok Yes Sync Ok Yes Link Active No
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
```

```
     752 bytes input
     240 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 667
1/2 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  7 bytes/sec
Five minute output rate 0 bytes/sec
     54332 bytes input
     1120 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/1 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 30 bytes/sec
     146390 bytes input
     217587 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/2 is DOWN Loopback No
Cable Length 50cm     Neighbor NONE
Link Ok Yes Sync Ok Yes Link Active No
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     1208 bytes input
     480 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/1 is OK Loopback No
Cable Length 50cm     Neighbor 1
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     41245 bytes input
     240 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/2 is OK Loopback No
Cable Length 50cm     Neighbor 2
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  10 bytes/sec
```

```
Five minute output rate 0 bytes/sec
     60412 bytes input
     480 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
```

Table 4: show switch stack-ports detail Command Output

Field	Description
Neighbor	Switch number of the active member at the other end of the stack cable.
Cable Length	Valid lengths are 50 cm, 1 m, or 3 m. If the switch cannot detect the cable length, the value is *Unknown*. The cable might not be connected, or the link might be unreliable.
Link OK	Whether the stack cable is connected and functional. There may or may not be a neighbor connected on the other end. The *link partner* is a stack port on a neighbor switch. • No: There is no stack cable connected to this port or the stack cable is not functional. • Yes: There is a functional stack cable connected to this port.
Link Active	Whether a neighbor is connected on the other end of the stack cable. • No: No neighbor is detected on the other end. The port cannot send traffic over this link. • Yes: A neighbor is detected on the other end. The port can send traffic over this link.
Sync OK	Whether the link partner sends valid protocol messages to the stack port. • No: The link partner does not send valid protocol messages to the stack port. • Yes: The link partner sends valid protocol messages to the port.
# Changes to LinkOK	The relative stability of the link. If a large number of changes occur in a short period of time, link flapping can occur.
Five minute input rate	The average rate (calculated over a five minute period) at which packets are received, measured in packets/sec.
Five minute output rate	The average rate (calculated over a five minute period) at which packets are transmitted, measured in packets/sec.

Field	Description
CRC Errors	Different types of Cyclic Redundancy Check (CRC) errors that are seen on a stack interface: • **Data CRC**: Stack interface data CRC error • **Ringword CRC**: Stack interface ring word CRC error • **InvRingWord**: Stack interface invalid ring word error • **PcsCodeWord**: Stack interface Physical Coding Sublayer (PCS) error These errors normally occur when a stack interface state changes due to a switchover or a switch reload. You can ignore such errors. But when these error counters increase significantly or when they increase continuously over a period of time, check the stack cable for issues.

he following is a sample output when the stack port flaps:

```
Device# show switch stack-ports detail
1/1 is OK Loopback No
Cable Length 50cm     Neighbor 2
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 4
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     320 bytes input
     80 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 770
1/2 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  5 bytes/sec
Five minute output rate 1 bytes/sec
     2949 bytes input
     320 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/1 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     49375 bytes input
     160 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/2 is OK Loopback No
```

```
Cable Length 50cm     Neighbor 1
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 2
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     1824 bytes input
     160 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/1 is OK Loopback No
Cable Length 50cm     Neighbor 1
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  372 bytes/sec
Five minute output rate 7 bytes/sec
     111876 bytes input
     4613 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/2 is OK Loopback No
Cable Length 50cm     Neighbor 2
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 2
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     80 bytes input
     0 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
```

The following is a sample output when a switch reloads:

```
Device#show switch stack-ports detail
1/1 is OK Loopback No
Cable Length 50cm     Neighbor 2
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 5
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     2032 bytes input
     320 bytes output
CRC Errors
            Data CRC 184
        Ringword CRC 187
        InvRingWord  120
         PcsCodeWord 112
1/2 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  2 bytes/sec
Five minute output rate 0 bytes/sec
     24164 bytes input
     800 bytes output
CRC Errors
            Data CRC 0
```

```
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/1 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     3024 bytes input
     240 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/2 is OK Loopback No
Cable Length 50cm     Neighbor 1
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  7 bytes/sec
Five minute output rate 0 bytes/sec
     9148 bytes input
     480 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/1 is OK Loopback No
Cable Length 50cm     Neighbor 1
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 15 bytes/sec
     1509354 bytes input
     27853 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/2 is OK Loopback No
Cable Length 50cm     Neighbor 2
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 3
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     240 bytes input
     160 bytes output
CRC Errors
            Data CRC 118
        Ringword CRC 74
        InvRingWord  125
         PcsCodeWord 373
```

Software Loopback: Examples

In a stack with three members, stack cables connect all the members:

```
# show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync   #Changes   In
```

```
          Status            Length    OK    Active  OK     To LinkOK  Loopback
--------  ------  --------  --------  ----  ------  ----  ---------  --------
  1/1     OK         3      50 cm     Yes    Yes    Yes        1        No
  1/2     OK         2      3 m       Yes    Yes    Yes        1        No
  2/1     OK         1      3 m       Yes    Yes    Yes        1        No
  2/2     OK         3      50 cm     Yes    Yes    Yes        1        No
  3/1     OK         2      50 cm     Yes    Yes    Yes        1        No
  3/2     OK         1      50 cm     Yes    Yes    Yes        1        No
```

If you disconnect the stack cable from Port 1 on Switch 1, these messages appear:

```
01:09:55: %STACKMGR-4-STACK_LINK_CHANGE: Stack Port 2 Switch 3 has changed to state DOWN
  01:09:56: %STACKMGR-4-STACK_LINK_CHANGE: Stack Port 1 Switch 1 has changed to state DOWN
```

```
  # show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback

  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     Absent    None    No cable  No     No     No         1        No
    1/2     OK         2      3 m       Yes    Yes    Yes        1        No
    2/1     OK         1      3 m       Yes    Yes    Yes        1        No
    2/2     OK         3      50 cm     Yes    Yes    Yes        1        No
    3/1     OK         2      50 cm     Yes    Yes    Yes        1        No
    3/2     Down      None    50 cm     No     No     No         1        No
```

If you disconnect the stack cable from Port 2 on Switch 1, the stack splits.

Switch 2 and Switch 3 are now in a two-member stack connected through stack cables:

```
# show sw stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    2/1     Down      None    3 m       No     No     No         1        No
    2/2     OK         3      50 cm     Yes    Yes    Yes        1        No
    3/1     OK         2      50 cm     Yes    Yes    Yes        1        No
    3/2     Down      None    50 cm     No     No     No         1        No
```

Switch 1 is a standalone switch:

```
# show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     Absent    None    No cable  No     No     No         1        Yes
    1/2     Absent    None    No cable  No     No     No         1        Yes
```

Software Loopback with Connected Stack Cables: Examples

- On Port 1 on Switch 1, the port status is *Down*, and a cable is connected.

  On Port 2 on Switch 1, the port status is *Absent*, and no cable is connected.

  ```
  # show switch stack-ports summary
    #
  ```

```
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     Down      None    50 Cm      No     No     No       1          No
    1/2     Absent    None    No cable   No     No     No       1          No
```

- In a *physical loopback*, a cable connects both stack ports on a switch. You can use this configuration to test
  - Cables on a switch that is running properly
  - Stack ports with a cable that works properly

```
# show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    2/1       OK       2      50 cm     Yes    Yes    Yes       1         No
    2/2       OK       2      50 cm     Yes    Yes    Yes       1         No
```

The port status shows that

- Switch 2 is a standalone switch.
- The ports can send and receive traffic.

Software Loopback with no Connected Stack Cable: Example

```
# show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     Absent    None    No cable   No     No     No       1         Yes
    1/2     Absent    None    No cable   No     No     No       1         Yes
```

Finding a Disconnected Stack Cable: Example

Stack cables connect all stack members. Port 2 on Switch 1 connects to Port 1 on Switch 2.

This is the port status for the members:

```
# show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     OK         2      50 cm     Yes    Yes    Yes       0         No
    1/2     OK         2      50 cm     Yes    Yes    Yes       0         No
    2/1     OK         1      50 cm     Yes    Yes    Yes       0         No
    2/2     OK         1      50 cm     Yes    Yes    Yes       0         No
```

If you disconnect the cable from Port 2 on Switch 1, these messages appear:

```
%STACKMGR-4-STACK_LINK_CHANGE: Stack Port 1 Switch 2 has changed to state DOWN

  %STACKMGR-4-STACK_LINK_CHANGE: Stack Port 2 Switch 1 has changed to state DOWN
```

This is now the port status:

```
  # show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1       OK       2      50 cm     Yes    Yes    Yes        1        No
    1/2     Absent    None    No cable   No     No     No        2        No
    2/1      Down     None    50 cm      No     No     No        2        No
    2/2       OK       1      50 cm     Yes    Yes    Yes        1        No
```

Only one end of the cable connects to a stack port, Port 1 on Switch 2.

- The *Stack Port Status* value for Port 2 on Switch 1 is *Absent*, and the value for Port 1 on Switch 2 is *Down*.
- The *Cable Length* value is *No cable*.

Diagnosing the problem:

- Verify the cable connection for Port 2 on Switch 1.
- Port 2 on Switch 1 has a port or cable problem if
  - The *In Loopback* value is *Yes*.

  or
  - The *Link OK*, *Link Active*, or *Sync OK* value is *No*.

Fixing a Bad Connection Between Stack Ports: Example

Stack cables connect all members. Port 2 on Switch 1 connects to Port 1 on Switch 2.

This is the port status:

```
# show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1       OK       2      50 cm     Yes    Yes    Yes        1          No
    1/2      Down     None    50 cm      No     No     No        2          No
    2/1      Down     None    50 cm      No     No     No        2          No
    2/2       OK       1      50 cm     Yes    Yes    Yes        1          No
```

Diagnosing the problem:

- The Stack Port Status value is *Down*.
- Link OK, Link Active, and Sync OK values are *No*.

- The Cable Length value is *50 cm*. The switch detects and correctly identifies the cable.

The connection between Port 2 on Switch 1 and Port 1 on Switch 2 is unreliable on at least one of the connector pins.

Additional References for Switch Stacks

Related Documents

Related Topic	Document Title
Cabling and powering on a switch stack.	https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9300/hardware/install/b_c9300_hig.html *Cisco Catalyst 9300 Series Switches Hardware Installation Guide*
SGACL High Availability	"Cisco TrustSec SGACL High Availability" module of the *Cisco TrustSec Switch Configuration Guide*

Error Message Decoder

Description	Link
To help you research and resolve system error messages in this release, use the Error Message Decoder tool.	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

Standards and RFCs

Standard/RFC	Title
None	—

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and , use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History for Switch Stacks

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	Switch Stack	A switch stack can have up to eight stacking-capable switches connected through their StackWise ports. The stack members work together as a unified system. Layer 2 and Layer 3 protocols present the entire switch stack as a single entity to the network.
Cisco IOS XE Amsterdam 17.3.1	Switch Stack	A new command **show switch stack-ports detail** was introduced to display detailed information on the stack link of each stack member.

Use the Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required