CONTENTS

CHAPTER 1
Managing Switch Stacks 1
Finding Feature Information 1
Prerequisites for Switch Stacks 1
Restrictions for Switch Stacks 1
Information About Switch Stacks 2
  Switch Stack Overview 2
  Switch Stack Membership 2
    Changes to Switch Stack Membership 3
Stack Member Numbers 3
Stack Member Priority Values 4
Switch Stack Bridge ID and MAC Address 5
  Persistent MAC Address on the Switch Stack 5
Active and Standby Switch Election and Reelection 5
Switch Stack Configuration Files 6
Offline Configuration to Provision a Stack Member 7
Upgrading a Switch Running Incompatible Software 7
Switch Stack Management Connectivity 7
How to Configure a Switch Stack 8
  Temporarily Disabling a Stack Port 8
  Reenabling a Stack Port While Another Member Starts 9
  Monitoring the Stack 9
Configuration Examples for Switch Stacks 10
  Switch Stack Configuration Scenarios 10
  Enabling the Persistent MAC Address Feature: Example 11
  Provisioning a New Member for a Switch Stack: Example 12
  show switch stack-ports summary Command Output: Example 12
System Mode Maintenance Counters 31
How to Configure Graceful Insertion and Removal 32
Creating a Maintenance Template 32
Configuring System Mode Maintenance 32
Starting and Stopping Maintenance Mode 33
Monitoring Graceful Insertion and Removal 34
Configuration Examples for Graceful Removal and Insertion 34
Example: Configuring Maintenance Templates 34
Example: Configuring System Mode Maintenance 35
Example: Starting and Stopping the Maintenance Mode 35
Example: Displaying System Mode Settings 35
Additional References for Graceful Insertion and Removal 36
Feature History and Information for Graceful Insertion and Removal 36

CHAPTER 4 Configuring 1:1 Redundancy 37
Prerequisites for 1:1 Redundancy 37
Information About 1:1 Redundancy 37
How to Configure 1:1 Redundancy 37
Enabling 1:1 Redundancy Stack Mode 37
Disabling 1:1 Redundancy Stack Mode 38
Configuration Examples for 1:1 Redundancy 38
Example: Enabling 1:1 Redundancy Stack Mode 38
Example: Disabling 1:1 Redundancy 39
Verifying the Stack Mode 39
Additional References for 1:1 Redundancy 39
Feature History for 1:1 Redundancy 39
CHAPTER 1

Managing Switch Stacks

• Finding Feature Information, on page 1
• Prerequisites for Switch Stacks, on page 1
• Restrictions for Switch Stacks, on page 1
• Information About Switch Stacks, on page 2
• How to Configure a Switch Stack, on page 8
• Configuration Examples for Switch Stacks, on page 10
• Feature History and Information for Switch Stacks, on page 18

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

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.
- 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 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 entities.
- 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.

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 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 joining a switch stack uses the system-level settings of that switch stack. If a is moved to a different switch stack before it is powered on, that loses its saved configuration file and uses the system-level configuration of the new switch stack. If the is powered on as a standalone before it joins the new switch stack, the stack will reload. When the stack reloads, the new 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. You do not need to reconfigure the interface settings. The
replacement (referred to as the provisioned) must have the same stack member number as the failed.

You back up and restore the stack configuration in the same way as you would for a standalone 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-adviser 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 supported network management applications such as CiscoWorks. 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.

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 switch stack-member-number stack port port-number disable</td>
<td>Disables the specified stack port.</td>
</tr>
<tr>
<td>Example: # switch 2 stack port 1 disable</td>
<td></td>
</tr>
<tr>
<td>Step 2 switch stack-member-number stack port port-number enable</td>
<td>Reenables the stack port.</td>
</tr>
<tr>
<td>Example: # switch 2 stack port 1 enable</td>
<td></td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Disconnect the stack cable between Port 1 on Switch 1 and Port 2 on Switch 4.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Remove Switch 4 from the stack.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Add a switch to replace Switch 4 and assign it switch-number 4.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Reconnect the cable between Port 1 on Switch 1 and Port 2 on Switch 4 (the replacement switch).</td>
</tr>
<tr>
<td>Step 5</td>
<td>Reenable the link between the switches. Enter the <code>switch 1 stack port 1 enable</code> privileged EXEC command to enable Port 1 on Switch 1.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Power on Switch 4.</td>
</tr>
</tbody>
</table>

**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 Stack

**Table 1: Commands for Displaying Stack Information**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show switch</code></td>
<td>Displays summary information about the stack, including the status of provisioned switches and switches in version-mismatch mode.</td>
</tr>
<tr>
<td><code>show switch stack-member-number</code></td>
<td>Displays information about a specific member.</td>
</tr>
<tr>
<td><code>show module</code></td>
<td>Displays summary information about the stack.</td>
</tr>
<tr>
<td><code>show switch detail</code></td>
<td>Displays detailed information about the stack.</td>
</tr>
<tr>
<td><code>show switch neighbors</code></td>
<td>Displays the stack neighbors.</td>
</tr>
<tr>
<td><code>show switch stack-ports [summary]</code></td>
<td>Displays port information for the stack. Use the <code>summary</code> keyword to display the stack cable length, the stack link status, and the loopback status.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>show redundancy</td>
<td>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.</td>
</tr>
<tr>
<td>show redundancy state</td>
<td>Displays all the redundancy states of the active and standby.</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active switch election specifically determined by existing active switches</td>
<td>Only one of the two active switches becomes the new active switch.</td>
</tr>
<tr>
<td>Connect two powered-on switch stacks through the StackWise ports.</td>
<td></td>
</tr>
<tr>
<td>Active switch election specifically determined by the stack member priority value</td>
<td>The stack member with the higher priority value is elected active switch.</td>
</tr>
<tr>
<td>1. Connect two switches through their StackWise ports.</td>
<td></td>
</tr>
<tr>
<td>2. Use the <strong>switch</strong> <code>stack-member-number priority new-priority-number</code> EXEC command to set one stack member with a higher member priority value.</td>
<td></td>
</tr>
<tr>
<td>3. Restart both member switches at the same time.</td>
<td></td>
</tr>
<tr>
<td>Active switch election specifically determined by the configuration file</td>
<td>The stack member with the saved configuration file is elected active switch.</td>
</tr>
<tr>
<td>Assuming that both member switches have the same priority value:</td>
<td></td>
</tr>
<tr>
<td>1. Make sure that one stack member has a default configuration and that the other stack member has a saved (nondefault) configuration file.</td>
<td></td>
</tr>
<tr>
<td>2. Restart both member switches at the same time.</td>
<td></td>
</tr>
<tr>
<td>Scenario</td>
<td>Result</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>Active switch election specifically determined by the MAC address</td>
<td>The stack member with the lower MAC address is elected active switch.</td>
</tr>
<tr>
<td>Stack member number conflict</td>
<td>The stack member with the higher priority value retains its stack member number. The other stack member has a new stack member number.</td>
</tr>
<tr>
<td>Add a stack member</td>
<td>The active switch is retained. The new switch is added to the switch stack.</td>
</tr>
<tr>
<td>Active switch failure</td>
<td>The standby switch becomes the new active switch. All other member switches in the stack remain as member switches and do not reboot.</td>
</tr>
<tr>
<td>Add eight member switches</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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:

```
(config)# stack-mac persistent timer 7
WARNING: The stack continues to use the base MAC of the old Master
WARNING: as the stack MAC after a master switchover until the MAC
WARNING: persistency timer expires. During this time the Network
WARNING: Administrators must make sure that the old stack-mac does
```
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 Status Length OK Active OK Changes Loopback
-------- ------ -------- -------- ---- ------ ---- --------- --------
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>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch#/Port#</td>
<td>Member number and its stack port number.</td>
</tr>
</tbody>
</table>
## Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack Port Status</td>
<td>Status of the stack port.</td>
</tr>
<tr>
<td></td>
<td>• Absent—No cable is detected on the stack port.</td>
</tr>
<tr>
<td></td>
<td>• Down—A cable is detected, but either no connected neighbor is up, or the stack port is disabled.</td>
</tr>
<tr>
<td></td>
<td>• OK—A cable is detected, and the connected neighbor is up.</td>
</tr>
<tr>
<td>Neighbor</td>
<td>Switch number of the active member at the other end of the stack cable.</td>
</tr>
<tr>
<td>Cable Length</td>
<td>Valid lengths are 50 cm, 1 m, or 3 m.</td>
</tr>
<tr>
<td></td>
<td>If the switch cannot detect the cable length, the value is <em>no cable</em>. The cable might not be connected, or the link might be unreliable.</td>
</tr>
<tr>
<td>Link OK</td>
<td>Whether the stack cable is connected and functional. There may or may not be a neighbor connected on the other end.</td>
</tr>
<tr>
<td></td>
<td>The <em>link partner</em> is a stack port on a neighbor switch.</td>
</tr>
<tr>
<td></td>
<td>• No—There is no stack cable connected to this port or the stack cable is not functional.</td>
</tr>
<tr>
<td></td>
<td>• Yes—There is a functional stack cable connected to this port.</td>
</tr>
<tr>
<td>Link Active</td>
<td>Whether a neighbor is connected on the other end of the stack cable.</td>
</tr>
<tr>
<td></td>
<td>• No—No neighbor is detected on the other end. The port cannot send traffic over this link.</td>
</tr>
<tr>
<td></td>
<td>• Yes—A neighbor is detected on the other end. The port can send traffic over this link.</td>
</tr>
<tr>
<td>Sync OK</td>
<td>Whether the link partner sends valid protocol messages to the stack port.</td>
</tr>
<tr>
<td></td>
<td>• No—The link partner does not send valid protocol messages to the stack port.</td>
</tr>
<tr>
<td></td>
<td>• Yes—The link partner sends valid protocol messages to the port.</td>
</tr>
<tr>
<td># Changes to LinkOK</td>
<td>The relative stability of the link.</td>
</tr>
<tr>
<td></td>
<td>If a large number of changes occur in a short period of time, link flapping can occur.</td>
</tr>
<tr>
<td>In Loopback</td>
<td>Whether a stack cable is attached to a stack port on the member.</td>
</tr>
<tr>
<td></td>
<td>• No—At least one stack port on the member has an attached stack cable.</td>
</tr>
<tr>
<td></td>
<td>• Yes—None of the stack ports on the member has an attached stack cable.</td>
</tr>
</tbody>
</table>

### Software Loopback: Examples

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

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:

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:

Switch 1 is a standalone switch:

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.
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

The port status shows that:

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

```shell
# 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:

```shell
# 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

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGACL High Availability</td>
<td>&quot;Cisco TrustSec SGACL High Availability&quot; module of the Cisco TrustSec Switch Configuration Guide</td>
</tr>
</tbody>
</table>

Error Message Decoder

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>To help you research and resolve system error messages in this release, use the Error Message Decoder tool.</td>
<td><a href="https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi">https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi</a></td>
</tr>
</tbody>
</table>

Standards and RFCs

<table>
<thead>
<tr>
<th>Standard/RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>All the supported MIBs for this release.</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS releases, and , use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>
Technical Assistance

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

Feature History and Information for Switch Stacks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 4: Feature Information for Switch Stacks

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch Stack</td>
<td>Cisco IOS XE Everest 16.5.1a</td>
<td>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.</td>
</tr>
</tbody>
</table>
Prerequisites for Cisco Nonstop Forwarding with Stateful Switchover

- Cisco nonstop forwarding (NSF) must be configured on a networking device that has been configured for stateful Switchover (SSO).
- Border Gateway Protocol (BGP) support in NSF requires that neighbor networking devices be NSF-aware; that is, devices must have the graceful restart capability and advertise that capability in their OPEN message during session establishment. If an NSF-capable device discovers that a particular BGP neighbor does not have graceful restart capability, it does not establish an NSF-capable session with that neighbor. All other neighbors that have graceful restart capability continue to have NSF-capable sessions with this NSF-capable networking device.
- Open Shortest Path First (OSPF) support in NSF requires that all neighbor networking devices be NSF-aware. If an NSF-capable device discovers that it has non-NSF-aware neighbors on a particular network segment, it disables NSF capabilities for that segment. Other network segments composed entirely of NSF-capable or NSF-aware devices continue to provide NSF capabilities.

Restrictions for Cisco Nonstop Forwarding with Stateful Switchover

The following are restrictions for configuring NSF with SSO:
Information About Cisco Nonstop Forwarding with Stateful Switchover

Overview of Cisco Nonstop Forwarding with Stateful Switchover

Cisco NSF works with the SSO feature. The device supports fault resistance by allowing a standby switch to take over if the active device becomes unavailable. NSF works with SSO to minimize the amount of time a network is unavailable.

Usually, when a networking device restarts, all routing peers of that device detect that the device went down and then came back up. This transition results in what is called a routing flap, which could spread across multiple routing domains. Routing flaps caused by routing restarts create routing instabilities, which are detrimental to the overall network performance. Cisco NSF helps to suppress routing flaps in SSO-enabled devices, thus reducing network instability.

Cisco NSF with SSO allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With NSF/SSO, peer networking devices do not experience routing flaps. Data traffic is forwarded through intelligent line cards or dual forwarding processors (FPs) while the standby router processor (RP) assumes control from the failed active RP during a switchover. NSF with SSO operation provides the ability of line cards and FPs to remain active through a switchover and to be kept current with the Forwarding Information Base (FIB) on the active RP.

NSF provides the following benefits:

• Improved network availability—NSF continues forwarding network traffic and application state information so that user session information is maintained after a switchover.

• Overall network stability—Network stability can be improved with the reduction in the number of route flaps that are created when devices in the network fail, and lose their routing tables.

• Neighboring devices do not detect a link flap—Because interfaces remain active during a switchover, neighboring devices do not detect a link flap (the link does not go down and come back up).

• Prevents routing flaps—Because SSO continues forwarding network traffic during a switchover, routing flaps are avoided.

• Maintains user sessions established prior to the switchover.
If a stack member does not respond, that member is removed from the stack.
If the standby device does not respond, a new standby device is elected.
If the active device does not respond, the standby device becomes the active device.

**SSO Operation**

When a standby device runs in SSO mode, the standby device starts up in a fully-initialized state and synchronizes with the persistent configuration and the running configuration on the active device. It subsequently maintains the state of the protocols, and all changes in hardware and software states for features that support SSO are kept in synchronization. Consequently, it offers minimum interruption to Layer 2 sessions in a redundant active device configuration.

If the active device fails, the standby device becomes the active device. This new active device uses existing Layer 2 switching information to continue forwarding traffic. Layer 3 forwarding is delayed until routing tables are repopulated in the newly active device.

**Cisco Nonstop Forwarding Operation**

NSF always runs with SSO, and provides redundancy for Layer 3 traffic. NSF is supported by BGP, Enhanced Interior Gateway Routing Protocol (EIGRP), and OSPF routing protocols and also by Cisco Express Forwarding for forwarding. These routing protocols have been enhanced with NSF-capability and awareness, which means that devices running these protocols can detect a switchover and take necessary actions to continue forwarding network traffic and to recover route information from peer devices.

Each protocol depends on Cisco Express Forwarding to continue forwarding packets during switchover while routing protocols rebuild the Routing Information Base (RIB) tables. After the convergence of routing protocols, Cisco Express Forwarding updates the FIB table and removes stale route entries. Cisco Express Forwarding then updates the hardware with the new FIB information.

If the active device is configured (with the `graceful-restart` command) for BGP, OSPF, or EIGRP routing protocols, routing updates are automatically sent during the active device election.

NSF has two primary components:

- **NSF-aware**: A networking device is NSF-aware if it is running NSF-compatible software. If neighboring devices detect that an NSF device can still forward packets when an active device election happens, this capability is referred to as NSF-awareness. Enhancements to the Layer 3 routing protocols (BGP, OSPF, and EIGRP) are designed to prevent route-flapping so that the Cisco Express Forwarding routing table does not time out or the NSF device does not drop routes. An NSF-aware device helps to send routing protocol information to the neighboring NSF device. NSF-awareness is enabled by default for EIGRP-stub, EIGRP, and OSPF protocols. NSF-awareness is disabled by default for BGP.

- **NSF-capability**: A device is NSF-capable if it is configured to support NSF; it rebuilds routing information from NSF-aware or NSF-capable neighbors. NSF works with SSO to minimize the amount of time that a Layer 3 network is unavailable following an active device election by continuing to forward IP packets. Reconvergence of Layer 3 routing protocols (BGP, OSPFv2, and EIGRP) is transparent to the user and happens automatically in the background. Routing protocols recover routing information from neighbor devices and rebuild the Cisco Express Forwarding table.
Cisco Express Forwarding

A key element of NSF is packet forwarding. In a Cisco networking device, packet forwarding is provided by Cisco Express Forwarding. Cisco Express Forwarding maintains the Forwarding Information Base (FIB), and uses the FIB information that is current at the time of a switchover to continue forwarding packets during a switchover, to reduce traffic interruption during the switchover.

During normal NSF operation, Cisco Express Forwarding on the active device synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the standby device. Upon switchover, the standby device initially has FIB and adjacency databases that are mirror images of those that were current on the active device. Cisco Express Forwarding keeps the forwarding engine on the standby device current with changes that are sent to it by Cisco Express Forwarding on the active device. The forwarding engine can continue forwarding after a switchover as soon as the interfaces and a data path are available.

As the routing protocols start to repopulate the RIB on a prefix-by-prefix basis, the updates cause prefix-by-prefix updates to Cisco Express Forwarding, which it uses to update the FIB and adjacency databases. Existing and new entries receive the new version (“epoch”) number, indicating that they have been refreshed. The forwarding information is updated on the forwarding engine during convergence. The device signals when the RIB has converged. The software removes all FIB and adjacency entries that have an epoch older than the current switchover epoch. The FIB now represents the newest routing protocol forwarding information.

Routing Protocols

Routing protocols run only on the active RP, and receive routing updates from neighbor devices. Routing protocols do not run on the standby RP. Following a switchover, routing protocols request that the NSF-aware neighbor devices send state information to help rebuild routing tables. Alternately, the Intermediate System-to-Intermediate System (IS-IS) protocol can be configured to synchronize state information from the active to the standby RP to help rebuild the routing table on the NSF-capable device in environments where neighbor devices are not NSF-aware.

Note

For NSF operation, routing protocols depend on Cisco Express Forwarding to continue forwarding packets while routing protocols rebuild the routing information.

BGP Operation

When a NSF-capable device begins a BGP session with a BGP peer, it sends an OPEN message to the peer. Included in the message is a declaration that the NSF-capable device has “graceful restart capability.” Graceful restart is the mechanism by which BGP routing peers avoid a routing flap following a switchover. If the BGP peer has this capability, it is aware that the device sending the message is NSF-capable. Both the NSF-capable device and its BGP peer(s) need to exchange the Graceful Restart Capability in their OPEN messages, at the time of session establishment. If both peers do not exchange the Graceful Restart Capability, the session is not graceful restart capable.

If the BGP session is lost during the RP switchover, the NSF-aware BGP peer marks all routes associated with the NSF-capable device as stale; however, it continues to use these routes to make forwarding decisions for a set period of time. This functionality means that no packets are lost while the newly active RP is waiting for convergence of the routing information with the BGP peers.
After an RP switchover occurs, the NSF-capable device reestablishes the session with the BGP peer. In establishing the new session, it sends a new graceful restart message that identifies the NSF-capable device as having restarted.

At this point, the routing information is exchanged between two BGP peers. Once this exchange is complete, the NSF-capable device uses the routing information to update the RIB and the FIB with the new forwarding information. The NSF-aware device uses the network information to remove stale routes from its BGP table. Following that, the BGP protocol is fully converged.

If a BGP peer does not support the graceful restart capability, it will ignore the graceful-restart capability in an OPEN message; but will establish a BGP session with the NSF-capable device. This function allows interoperability with non-NSF-aware BGP peers (and without NSF functionality), but the BGP session with non-NSF-aware BGP peers will not be graceful restart capable.

---

**Note**

BGP support in NSF requires that neighbor networking devices be NSF-aware; that is, devices must have the Graceful Restart Capability and advertise that capability in their OPEN message during session establishment. If an NSF-capable device discovers that a particular BGP neighbor does not have Graceful Restart Capability, it will not establish an NSF-capable session with that neighbor. All other neighbors that have Graceful Restart Capability will continue to have NSF-capable sessions with this NSF-capable networking device.

---

**EIGRP Operation**

Enhanced Interior Gateway Routing Protocol (EIGRP) NSF capabilities are exchanged by EIGRP peers in hello packets. The NSF-capable device notifies its neighbors that an NSF restart operation has started by setting the restart (RS) bit in a hello packet. When an NSF-aware device receives notification from an NSF-capable neighbor that an NSF-restart operation is in progress, the NSF-capable and NSF-aware devices immediately exchange their topology tables. The NSF-aware device sends an end-of-table update packet when the transmission of its topology table is complete. The NSF-aware device then performs the following actions to assist the NSF-capable device:

- The EIGRP hello hold timer is expired to reduce the time interval set for hello packet generation and transmission. This allows the NSF-aware device to reply to the NSF-capable device more quickly reducing the amount of time required for the NSF-capable device to rediscover neighbors and rebuild the topology table.

- The route-hold timer is started. This timer is used to set the period of time that the NSF-aware device will hold known routes for the NSF-capable neighbor. This timer is configured with the `timers nsf route-hold` command. The default time period is 240 seconds.

- In the peer list, the NSF-aware device notes that the NSF-capable neighbor is restarting, maintains adjacency, and holds known routes for the NSF-capable neighbor until the neighbor signals that it is ready for the NSF-aware device to send its topology table, or the route-hold timer expires. If the route-hold timer expires on the NSF-aware device, the NSF-aware device discards held routes and treats the NSF-capable device as a new device joining the network and reestablishes adjacency accordingly.

- The NSF-aware device continues to send queries to the NSF-capable device which is still in the process of converging after a switchover, effectively extending the time before a stuck-in-active condition can occur.

When the switchover operation is complete, the NSF-capable device notifies its neighbors that it has reconverged and has received all of their topology tables by sending an end-of-table update packet to assisting devices. The NSF-capable device then returns to normal operation. The NSF-aware device will look for alternate paths...
(go active) for any routes that are not refreshed by the NSF-capable (restarting device). The NSF-aware device will then return to normal operation. If all paths are refreshed by the NSF-capable device, the NSF-aware device will immediately return to normal operation.

---

**Note**

NSF-aware devices are completely compatible with non-NSF aware or -capable neighbors in an EIGRP network. A non-NSF aware neighbor will ignore NSF capabilities and reset adjacencies and otherwise maintain the peering sessions normally.

---

**OSPF Operation**

When an OSPF NSF-capable device performs a supervisor engine switchover, it must perform the following tasks in order to resynchronize its link state database with its OSPF neighbors:

- Relearn the available OSPF neighbors on the network without causing a reset of the neighbor relationship.
- Reacquire the contents of the link state database for the network.

As quickly as possible after a supervisor engine switchover, the NSF-capable device sends an OSPF NSF signal to neighboring NSF-aware devices. Neighbor networking devices recognize this signal as an indicator that the neighbor relationship with this device should not be reset. As the NSF-capable device receives signals from other devices on the network, it can begin to rebuild its neighbor list.

After neighbor relationships are reestablished, the NSF-capable device begins to resynchronize its database with all of its NSF-aware neighbors. At this point, the routing information is exchanged between the OSPF neighbors. Once this exchange is complete, the NSF-capable device uses the routing information to remove stale routes, update the RIB, and update the FIB with the new forwarding information. The OSPF protocols are then fully converged.

---

**Note**

OSPF support in NSF requires that all neighbor networking devices be NSF-aware. If an NSF-capable device discovers that it has non-NSF-aware neighbors on a particular network segment, it disables NSF capabilities for that segment. Other network segments composed entirely of NSF-capable or NSF-aware devices continue to provide NSF capabilities.

---

**How to Configure Cisco Nonstop Forwarding with Stateful Switchover**

**Configuring Stateful Switchover**

You must configure SSO in order to use NSF with any supported protocol.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> show redundancy states</td>
<td>Displays the operating redundancy mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show redundancy states</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> redundancy</td>
<td>Enters redundancy configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# redundancy</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> mode sso</td>
<td>Configures stateful switchover.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-red)# mode sso</td>
<td>• When this command is entered, the standby switch is reloaded and begins to work in SSO mode.</td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Exits redundancy configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-red)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> show redundancy states</td>
<td>Displays the operating redundancy mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show redundancy states</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> debug redundancy status</td>
<td>Enables the debugging of redundancy status events.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# debug redundancy status</td>
<td></td>
</tr>
</tbody>
</table>

## Verifying Cisco Express Forwarding with Cisco Nonstop Forwarding

### Procedure

**show cef state**

Displays the state of Cisco Express Forwarding on a networking device.

**Example:**

Device# show cef state

CEF Status:

RF instance

common CEF enabled
IPv4 CEF Status:
CEF enabled/running
dCEF enabled/running
CEF switching enabled/running
universal per-destination load sharing algorithm, id DEA83012
IPv6 CEF Status:
CEF disabled/not running
dCEF disabled/not running
universal per-destination load sharing algorithm, id DEA83012
RRP state:
I am standby RRP: no
RF Peer Presence: yes
RF PeerComm reached: yes
RF Progression blocked: never
Redundancy mode: rpr(1)
CEF NSF sync: disabled/not running
CEF ISSU Status:
FIBHWIDB broker
No slots are ISSU capable.
FIBIDB broker
No slots are ISSU capable.
FIBHWIDB Subblock broker
No slots are ISSU capable.
FIBIDB Subblock broker
No slots are ISSU capable.
Adjacency update
No slots are ISSU capable.
IPv4 table broker
No slots are ISSU capable.
CEF push
No slots are ISSU capable.

Configuration Examples for Cisco Nonstop Forwarding with Stateful Switchover

Example: Configuring Stateful Switchover

This example shows how to configure the system for SSO and displays the redundancy state:

Device(config)# redundancy
Device(config-red)# mode sso
Device(config-red)# end
Device#

The following is sample output from the `show redundancy states` command:

```
show redundancy states
  my state = 13 -ACTIVE
  peer state = 8 -STANDBY HOT
  Mode = Duplex
  Unit = Primary
  Unit ID = 5
  Redundancy Mode (Operational) = sso
  Redundancy Mode (Configured) = sso
```
Additional References for Nonstop Forwarding with Stateful Switchover

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>For complete syntax and usage information for the commands used in this chapter.</td>
<td>See the Stack Manager and High Availability section of the Command Reference (Catalyst 9300 Series Switches)</td>
</tr>
</tbody>
</table>

Feature History and Information for Cisco Nonstop Forwarding with Stateful Switchover

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonstop Forwarding with Stateful Switchover</td>
<td>Cisco IOS XE Everest 16.6.1</td>
<td>This feature was introduced. Cisco NSF works with the SSO feature. NSF works with SSO to minimize the amount of time a network is unavailable to users following a switchover. The main objective of NSF SSO is to continue forwarding IP packets following a Route Processor (RP) switchover.</td>
</tr>
</tbody>
</table>
Configuring NSF with SSO

Feature History and Information for Cisco Nonstop Forwarding with Stateful Switchover
CHAPTER 3

Configuring Graceful Insertion and Removal

Graceful Insertion and Removal (GIR) provides an alternative method to minimize network service impact caused by device maintenance. GIR leverages redundant paths in the network to smoothly remove a device under maintenance, out of service, and insert it back to service when the maintenance is complete. This module describes the how to configure GIR.

- Restrictions for Graceful Insertion and Removal, on page 29
- Information About Graceful Insertion and Removal, on page 29
- How to Configure Graceful Insertion and Removal, on page 32
- Monitoring Graceful Insertion and Removal, on page 34
- Configuration Examples for Graceful Removal and Insertion, on page 34
- Additional References for Graceful Insertion and Removal, on page 36
- Feature History and Information for Graceful Insertion and Removal, on page 36

Restrictions for Graceful Insertion and Removal

GIR is supported for layer two interface shutdown, ISIS routing protocol, HSRP, VRRPv3 and BGP. This is configured either by creating customized templates or without a template.

Information About Graceful Insertion and Removal

Overview

Graceful Insertion and Removal (GIR) isolates a switch from the network in order to perform debugging or an upgrade. The switch can be put into maintenance mode using the start maintenance command. When switch maintenance is complete, the switch will return to normal mode on either reaching the configured maintenance timeout, or by enabling the stop maintenance command.

Creating a maintenance mode template before you put the switch in maintenance mode is optional. The objective of maintenance mode for a device is to minimize traffic disruption at the time of removal from the network, as well as during the time of insertion. There are mainly three stages:

- Graceful removal of the node from network.
- Performing maintenance on the device.
• Graceful insertion into the network.

A switch can be put into maintenance mode using default template or a custom template. The default template contains all the ISIS instances, along with `shutdown l2`. In the custom template, you can configure the required ISIS instances and `shutdown l2` option. On entering maintenance mode, all participating protocols are isolated, and L2 ports are shut down. When normal mode is restored, all the protocols and L2 ports are brought back up.

Snapshots are taken automatically while entering and exiting the maintenance mode. You can use the `snapshot create snapshot-name snapshot-description` command to capture and store snapshots for pre-selected features. Snapshots are useful to compare the state of a switch before it went into maintenance mode and after it came back to normal mode. The snapshot process consists of three parts:

• Creating a snapshot of the states of a few preselected features on the switch and storing them on the persistent storage media.

• Listing the snapshots taken at various time intervals and managing them.

• Comparing snapshots and showing the summary and details of each feature.

The maximum number of snapshots that may be stored on the switch is 10. You can use the `snapshot delete snapshot-name` command, to delete a specific snapshot from the device.

You can create multiple templates for the maintenance template or the snapshot template. But only one maintenance template and one snapshot template can be applied to the device at one time.

Snapshot templates can be created to generate specific snapshots. A new snapshot template can be created using the `snapshot-template template-name` command. The command `snapshot-template default-snapshot-template` can be used to specify the default snapshot template in the maintenance mode. The `snapshot create [template template-name] snapshot-name snapshot-description` command can be used to apply a specific template to the snapshot create feature.

### Layer 2 Interface Shutdown

Layer 2 interfaces, such as ports on a switch, are shut down when the system is transitioning into maintenance mode. Layer 2 interfaces are shut down by using the `shutdown l2` (maintenance template configuration mode) command in the custom template.

### Custom Template

As a network administrator, you can create a template that is applied when the system goes into maintenance mode. This allows you to isolate specific protocols. All instances that need to be isolated must be explicitly specified.

You can create multiple templates with different configurations. However, only a single template is applied to the maintenance mode CLI. Once applied, the template cannot be updated. If the template has to be updated, then you must remove it, make the changes, and then re-apply.

Within a template, protocols belonging to one class are serviced in parallel. The order of priority of the protocols is the same as that of the default template.

To configure this feature, enter the maintenance mode using the `system mode maintenance` command and enable the feature using the `template template-name class` command.

For example if the custom template has the following protocols:
Maintenance-template foo
   router isis 100
   hsrp Et0/1 1
   hsrp Et0/1 2
   router isis 200
Maintenance-template foo class
   router isis 100
   hsrp Et0/1 1
   hsrp Et0/1 2
   router isis 200

In the above example, since isis belongs to CLASS_I, router isis 100 & router isis 200 will be serviced in parallel. Once acknowledgements are received for both these protocols belonging to IGP class, FHRP_CLASS clients, hsrp Et0/1 and hsrp Et0/1 2 will be serviced in parallel.

When the template-class feature is configured, the protocols follow an order based on the class they belong to when entering maintenance mode. The protocols follow the opposite order when returning to normal mode.

**System Mode Maintenance Counters**

GIR has counters to track the following events:

- Number of times the switch went into maintenance.
- Ack statistics per client.
- Nack statistics per client
- Number of times a particular client did not acknowledge.
- Number of times switch over happened during GIR. GIR infra will rsync this counter to track multiple switchovers.
- Number of times the failsafe timer expired.
- Number of times system got out of maintenance on a timeout expiry.

Enter the `show system mode maintenance counters` command in privileged EXEC mode, to display the counters that are being tracked by the feature.

Enter the `clear system mode maintenance counters` command in privileged EXEC mode, to clear the counters supported by the feature.

The client-ack timeout value can be configured using the `failsafe failsafe-timeout-value` command. Failsafe time is the time that the GIR engine allows a client to transition. Each client sends a notification to the GIR engine about its transition. If it takes more than the failsafe time to transition, it is assumed to have transitioned. The failsafe timer can be configured between 5 - 180 minutes, with a default of 30 minutes.
# How to Configure Graceful Insertion and Removal

## Creating a Maintenance Template

### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enable</td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>maintenance-template template_name</td>
<td>Creates a template with the specified name. For example, see Examples: Creating customer profile.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config)# maintenance-template gir1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>router routing_protocol instance_id</td>
<td>Creates instances that should be isolated under this template.</td>
</tr>
<tr>
<td></td>
<td>shutdown l2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-maintenance-templ)# router isis 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-maintenance-templ)# shutdown 12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-maintenance-templ)# router bgp AS-number</td>
<td></td>
</tr>
</tbody>
</table>

## Configuring System Mode Maintenance

### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enable</td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>
### Starting and Stopping Maintenance Mode

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters system mode maintenance configuration mode.</td>
</tr>
<tr>
<td>system mode maintenance</td>
<td>Different sub commands to create maintenance mode parameters are configured in this mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# system mode maintenance</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Configures maintenance mode parameters.</td>
</tr>
<tr>
<td>timeout timeout-value</td>
<td>Configures maintenance mode timeout period in minutes, after which the system automatically returns to normal mode. The default timeout value is never.</td>
</tr>
<tr>
<td>template template-name</td>
<td>Configures maintenance mode using the specified template.</td>
</tr>
<tr>
<td>failsafe fail-safe-timeout-value</td>
<td>Configures client-ack timeout value.</td>
</tr>
<tr>
<td>on-reload reset-reason maintenance</td>
<td>If the system is going into maintenance mode, it will continue to reach maintenance. If the system is exiting from maintenance mode, then it will reach normal mode.</td>
</tr>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Puts the system into maintenance mode.</td>
</tr>
<tr>
<td>start maintenance</td>
<td>Puts the system back into normal mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device# start maintenance</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Puts the system into maintenance mode.</td>
</tr>
<tr>
<td>stop maintenance</td>
<td>Puts the system back into normal mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device# stop maintenance</td>
</tr>
</tbody>
</table>
Monitoring Graceful Insertion and Removal

Use the following commands to check the status of or display statistics generated by the GIR feature:

**Table 6: Privileged EXEC Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show system mode maintenance template-name</code></td>
<td>Displays information about system mode.</td>
</tr>
<tr>
<td><code>show system snapshots dump &lt;snapshot-file-name&gt;</code></td>
<td>Displays all the snapshots present on the device.</td>
</tr>
<tr>
<td><code>show system snapshots dump &lt;snapshot-file-name&gt; xml</code></td>
<td>Displays all the snapshots present on the device in XML format.</td>
</tr>
<tr>
<td><code>show system snapshots compare snapshot-name1 snapshot-name2</code></td>
<td>Displays differences between snapshots taken before entering maintenance mode and after exiting from the maintenance mode.</td>
</tr>
</tbody>
</table>

**Table 7: Global Configuration Commands for Troubleshooting**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>debug system mode maintenance</code></td>
<td>Displays information to help troubleshoot the GIR feature.</td>
</tr>
</tbody>
</table>

Configuration Examples for Graceful Removal and Insertion

The following examples show the sequence followed to enable GIR during a maintenance window.

**Example: Configuring Maintenance Templates**

Any protocol that is supported by GIR can be configured in the maintenance template. This example shows how to configure a maintenance template t1 with an ISIS routing protocol instance.

```
Device# configure terminal
Device(config)# maintenance-template t1
Device(config-maintenance-templ)# router isis 1
```

This example shows how to configure a maintenance template t1 with shutdown l2.

```
Device# configure terminal
Device(config)# maintenance-template t1
Device(config-maintenance-templ)# shutdown l2
```

This example shows how to configure a maintenance template t1 with a BGP routing protocol instance.
Example: Configuring System Mode Maintenance

This example shows how to create a maintenance template and configure the maintenance mode parameters.

```
Device# configure terminal
Device(config)# maintenance-template t1
```

```
Device(config-maintenance-templ)# router BGP 1
```

Example: Starting and Stopping the Maintenance Mode

This example shows how to put the system into maintenance mode.

```
Device# start maintenance
```

After the activity is completed, the system can be put out of maintenance mode.

This example shows how to put the system out of maintenance mode.

```
Device# stop maintenance
```

Example: Displaying System Mode Settings

This example shows how to display system mode settings using different options.

```
Device# show system mode
System Mode: Normal
```

```
Device# show system mode maintenance
System Mode: Normal
Current Maintenance Parameters:
  Maintenance Duration: 15(mins)
  Failsafe Timeout: 30(mins)
  Maintenance Template: t1
  Reload in Maintenance: False
```

```
Device# show system mode maintenance clients
System Mode: Normal
Maintenance Clients:
  CLASS-EGP
  CLASS-IGP
  router isis 1: Transition None
  CLASS-MCAST
  CLASS-L2
```

```
Device# show system mode maintenance template default
System Mode: Normal
default maintenance-template details:
```
**Additional References for Graceful Insertion and Removal**

**Related Documents**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>For complete syntax and usage information for the commands used in this chapter.</td>
<td>See the <em>Stack Manager and High Availability</em> section of the <em>Command Reference (Catalyst 9300 Series Switches)</em></td>
</tr>
</tbody>
</table>

**Feature History and Information for Graceful Insertion and Removal**

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graceful Insertion and Removal</td>
<td>Cisco IOS XE Everest 16.6.1</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>
| Support for snapshot templates and additional protocols | Cisco IOS XE Fuji 16.9.1 | The following enhancements have been added to the GIR feature:  
  • Snapshot templates can be used to generate specific snapshots. Protocols belonging to one class within the same custom template will be serviced in parallel. System mode maintenance counters have been added to track several events such as the number of times the switch went into maintenance.  
  • GIR is now supported for the HSRP protocol  
  • GIR is now supported for VRRPv3 protocol. |
| Support for BGP | Cisco IOS XE Gibraltar 16.10.1 | GIR is now supported for BGP. |
CHAPTER 4

Configuring 1:1 Redundancy

• Prerequisites for 1:1 Redundancy, on page 37
• Information About 1:1 Redundancy, on page 37
• How to Configure 1:1 Redundancy, on page 37
• Configuration Examples for 1:1 Redundancy, on page 38
• Verifying the Stack Mode, on page 39
• Additional References for 1:1 Redundancy, on page 39
• Feature History for 1:1 Redundancy, on page 39

Prerequisites for 1:1 Redundancy

• 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 Configuration Guide of the required release.

• All the switches in the stack must be running compatible software versions.

Information About 1:1 Redundancy

1:1 redundancy is used to assign active and standby roles to specific switches in the stack. This overrides the traditional N+1 role selection algorithm, where any switch in the stack can be active or standby. In 1:1 redundancy, the stack manager determines the active and standby role for a specific switch, based on the flash ROMMON variable. The algorithm assigns one switch as active, another switch as standby, designating all remaining switches in the stack as members. When an active switch reboots it becomes standby and the existing standby switch becomes the new active. The existing member switches remain in the same state.

How to Configure 1:1 Redundancy

Enabling 1:1 Redundancy Stack Mode

Follow these steps to enable the 1:1 redundancy stack mode, and set a switch as the active switch in a stack, or as the standby:
Disabling 1:1 Redundancy Stack Mode

On a switch where 1:1 redundancy is enabled, follow these steps to disable the feature. This changes the stack mode to N+1:

### Disabling 1:1 Redundancy Stack Mode

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device&gt; enable</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 2</strong></th>
<th>Changes stack mode to 1:1 mode and designates the switch as active or standby.</th>
</tr>
</thead>
<tbody>
<tr>
<td>`switch switch-number role {active</td>
<td>standby}`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device# switch 1 role active</code></td>
<td></td>
</tr>
</tbody>
</table>

### Configuration Examples for 1:1 Redundancy

**Example: Enabling 1:1 Redundancy Stack Mode**

You can use the `switch switch-number role` command to set the active and standby switch in 1:1 stack mode. The stack operates in the 1:1 stack mode with the specified active or standby after reboot. In the following example, switch 1 is assigned the active role, and switch 2 is assigned the standby role.

**Device# switch 1 role active**

**WARNING:** Changing the switch role may result in redundancy mode being configured to 1+1 mode for this stack. If the configured Active or Standby switch numbers do not boot up, then the stack will not be able to boot. Do you want to continue? [y/n]? [yes]: yes

**Device# switch 2 role standby**

**WARNING:** Changing the switch role may result in redundancy mode being configured to 1+1 mode.
mode for this stack. If the configured Active or Standby switch numbers do not boot up, then the stack will not be able to boot. Do you want to continue? [y/n]? [yes]: yes

Example: Disabling 1:1 Redundancy

You can use the `switch clear stack-mode` command to remove 1:1 stack mode, and change it back to N+1 stack mode.

```
Device# switch clear stack-mode
```

WARNING: Clearing the chassis HA configuration will result in the chassis coming up in Stand Alone mode after reboot. The HA configuration will remain the same on other chassis. Do you wish to continue? [y/n]? [yes]:

Verifying the Stack Mode

To verify the current stack mode on a switch, enter the `show switch stack-mode` command in privileged EXEC mode. The output displays detailed status of the currently running stack mode.

```
Device# show switch stack-mode
```

<table>
<thead>
<tr>
<th>Switch</th>
<th>Role</th>
<th>Mac Address</th>
<th>Version</th>
<th>Mode</th>
<th>Configured</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Member</td>
<td>3c5e.c357.c880</td>
<td>1+1'</td>
<td>Active'</td>
<td>Ready</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Active</td>
<td>547c.69de.cd00</td>
<td>V05</td>
<td>1+1'</td>
<td>Standby'</td>
<td>Ready</td>
</tr>
<tr>
<td>3</td>
<td>Member</td>
<td>547c.6965.cf80</td>
<td>V05</td>
<td>1+1'</td>
<td>Member'</td>
<td>Ready</td>
</tr>
</tbody>
</table>

The `Mode` field indicates the current stack mode.

The `Configured` field refers to the switch state expected after a reboot.

Single quotation marks (') indicate that the stack mode has been changed.

Additional References for 1:1 Redundancy

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>For complete syntax and usage information for the commands used in this chapter.</td>
<td>See the Stacking and High Availability Commands section of the Command Reference Command Reference for the release.</td>
</tr>
</tbody>
</table>

Feature History for 1:1 Redundancy

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.
### Feature History for 1:1 Redundancy

<table>
<thead>
<tr>
<th>Release</th>
<th>Feature</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.6.1</td>
<td>1:1 Redundancy</td>
<td>Enables the 1:1 redundancy stack mode and enables you to assign active and standby roles to specific switches in the stack.</td>
</tr>
</tbody>
</table>

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [http://www.cisco.com/go/cfn](http://www.cisco.com/go/cfn).