High Availability Configuration Guide, Cisco IOS XE Bengaluru 17.4.x (Catalyst 9400 Switches)

First Published: 2020-11-30

Last Modified: 2023-06-13

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2020 Cisco Systems, Inc. All rights reserved.

CONTENTS

CHAPTER 1 **Configuring Nonstop Forwarding with Stateful Switchover** 1

Prerequisites for Cisco Nonstop Forwarding with Stateful Switchover 1

Restrictions for Cisco Nonstop Forwarding with Stateful Switchover 2

Information About Cisco Nonstop Forwarding with Stateful Switchover 2

Overview of Cisco Nonstop Forwarding with Stateful Switchover 2

SSO Operation 3

Cisco Nonstop Forwarding Operation 3

Cisco Express Forwarding 4

Routing Protocols 4

BGP Operation 5

EIGRP Operation 5

OSPF Operation 6

How to Configure Cisco Nonstop Forwarding with Stateful Switchover 7

Configuring Stateful Switchover 7

Verifying Cisco Express Forwarding with Cisco Nonstop Forwarding 8

Configuration Examples for Cisco Nonstop Forwarding with Stateful Switchover 9

Example: Configuring Stateful Switchover 9

Additional References for Cisco Nonstop Forwarding with Stateful Switchover 11

Feature History for Cisco Nonstop Forwarding with Stateful Switchover 11

CHAPTER 2 **Configuring Graceful Insertion and Removal** 13

Restrictions for Graceful Insertion and Removal 13

Information About Graceful Insertion and Removal 13

Overview 13

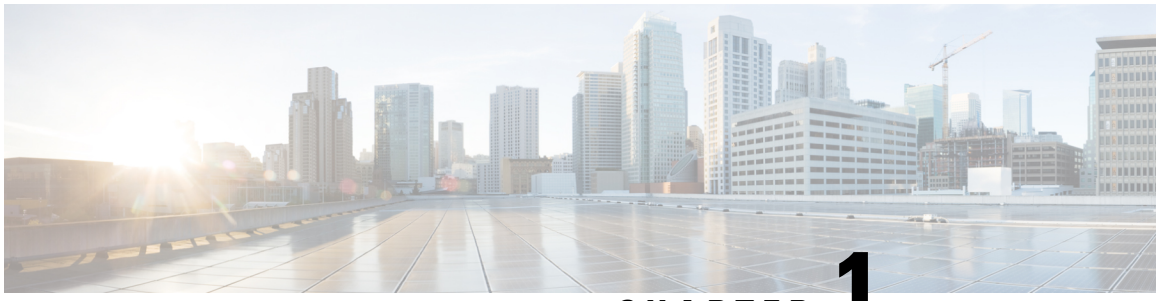
Layer 2 Interface Shutdown 14

Custom Template 14

System Mode Maintenance Counters 15
How to Configure Graceful Insertion and Removal 16
Creating a Maintenance Template 16
Configuring System Mode Maintenance 16
Starting and Stopping Maintenance Mode 17
Monitoring Graceful Insertion and Removal 18
Configuration Examples for Graceful Removal and Insertion 18
Example: Configuring Maintenance Templates 18
Example: Configuring System Mode Maintenance 19
Example: Starting and Stopping the Maintenance Mode 19
Example: Displaying System Mode Settings 19
Additional References for Graceful Insertion and Removal 20
Feature History for Graceful Insertion and Removal 20

CHAPTER 3 **Configuring Cisco StackWise Virtual 23**
Prerequisites for Cisco StackWise Virtual 23
Restrictions for Cisco StackWise Virtual 24
Information About Cisco StackWise Virtual 25
Cisco StackWise Virtual on Cisco Catalyst 9400 Series Switches 25
Overview of Cisco StackWise Virtual 27
Cisco StackWise Virtual Topology 27
Cisco StackWise Virtual Redundancy 29
SSO Redundancy 29
Nonstop Forwarding 30
Multichassis EtherChannels 30
MEC Minimum Latency Load Balancing 30
MEC Failure Scenarios 31
Cisco StackWise Virtual Packet Handling 31
Traffic on StackWise Virtual link 31
Layer 2 Protocols 32
Layer 3 Protocols 34
Dual-Active Detection 36
Dual-Active-Detection Link with Fast Hello 36
Dual-Active Detection with enhanced PAgP 36

Recovery Actions 37
Implementing Cisco StackWise Virtual 37
How to Configure Cisco StackWise Virtual 38
Configuring Cisco StackWise Virtual Settings 38
Configuring Cisco StackWise Virtual Link 39
Configuring BUM Traffic Optimization 41
Configuring StackWise Virtual Fast Hello Dual-Active-Detection Link 41
Enabling ePAgP Dual-Active-Detection 42
Disabling Recovery Reload 45
Disabling Cisco StackWise Virtual 45
Configuration Examples for StackWise Virtual 47
Example: Configuring StackWise Virtual Link 47
Example: Configuring StackWise Virtual Fast Hello Dual-Active-Detection Link 48
Example: Displaying StackWise Virtual Link Information 48
Example: Displaying StackWise Virtual Dual-Active-Detection Link Information 48
Verifying Cisco StackWise Virtual Configuration 49
Additional References for StackWise Virtual 50
Feature History for Cisco StackWise Virtual 50



CHAPTER 1

Configuring Nonstop Forwarding with Stateful Switchover

Cisco Nonstop Forwarding (NSF) works with the Stateful Switchover (SSO) feature 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.

- Prerequisites for Cisco Nonstop Forwarding with Stateful Switchover, on page 1
- Restrictions for Cisco Nonstop Forwarding with Stateful Switchover, on page 2
- Information About Cisco Nonstop Forwarding with Stateful Switchover, on page 2
- How to Configure Cisco Nonstop Forwarding with Stateful Switchover, on page 7
- Verifying Cisco Express Forwarding with Cisco Nonstop Forwarding, on page 8
- Configuration Examples for Cisco Nonstop Forwarding with Stateful Switchover, on page 9
- Additional References for Cisco Nonstop Forwarding with Stateful Switchover, on page 11
- Feature History for Cisco Nonstop Forwarding with Stateful Switchover, on page 11

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:

- For NSF operation, you must have SSO configured on the device.
- All Layer 3 neighboring devices must be an NSF helper or NSF-capable to support graceful restart capability.
- For IETF, all neighboring devices must be running an NSF-aware software image.
- The Hot Standby Routing Protocol (HSRP) is not supported with NSF SSO.
- An NSF-aware device cannot support two NSF-capable peers performing an NSF restart operation at the same time. However, both neighbors can reestablish peering sessions after the NSF restart operation is complete.
- For SSO operation, ensure that both active and standby devices run the same version of the Cisco IOS XE image. If the active and standby devices are operating different images, SSO failover might cause an outage.

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.

Note

- If the standby device is not programmed with a Federal Information Processing Standards (FIPS) key, it will print warning messages since it is not in the correct operating mode.
- The switch will work in SSO mode even if one supervisor module is in FIPS mode and the other, in non-FIPS mode.

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

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

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:
RP 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** command:

```
Device# show redundancy states

       my state = 13 -ACTIVE
     peer state = 1  -DISABLED
           Mode = Simplex
           Unit = Primary
        Unit ID = 3

Redundancy Mode (Operational) = Non-redundant
Redundancy Mode (Configured)  = sso
Redundancy State              = Non Redundant
     Maintenance Mode = Disabled
    Manual Swact = disabled (system is simplex (no peer unit))
 Communications = Down      Reason: Simplex mode

   client count = 103
 client_notification_TMR = 30000 milliseconds
           RF debug mask = 0x0
```

The following is sample output from the **show redundancy clients** command:

```
Device# show redundancy clients

 clientID = 29       group_id = 1        clientSeq = 60       Redundancy Mode RF
 clientID = 139      group_id = 1        clientSeq = 62       IfIndex
 clientID = 25       group_id = 1        clientSeq = 71       CHKPT RF
 clientID = 10001    group_id = 1        clientSeq = 85       QEMU Platform RF
 clientID = 77       group_id = 1        clientSeq = 87       Event Manager
 clientID = 1340     group_id = 1        clientSeq = 104      RP Platform RF
 clientID = 1501     group_id = 1        clientSeq = 105      CWAN HA
 clientID = 305      group_id = 1        clientSeq = 110      Multicast ISSU Consolidation
RF
 clientID = 304      group_id = 1        clientSeq = 111      IP multicast RF Client
 clientID = 22       group_id = 1        clientSeq = 112      Network RF Client
 clientID = 88       group_id = 1        clientSeq = 113      HSRP
 clientID = 114      group_id = 1        clientSeq = 114      GLBP
 clientID = 4700     group_id = 1        clientSeq = 118      COND_DEBUG RF
 clientID = 1341     group_id = 1        clientSeq = 119      IOSXE DPIDX
 clientID = 1505     group_id = 1        clientSeq = 120      IOSXE SPA TSM
 clientID = 75       group_id = 1        clientSeq = 130      Tableid HA
 clientID = 501      group_id = 1        clientSeq = 137      LAN-Switch VTP VLAN
 clientID = 71       group_id = 1        clientSeq = 139      XDR RRP RF Client
 clientID = 24       group_id = 1        clientSeq = 140      CEF RRP RF Client
```

```
clientID = 146      group_id = 1        clientSeq = 142      BFD RF Client
clientID = 301      group_id = 1        clientSeq = 146      MRIB RP RF Client
clientID = 306      group_id = 1        clientSeq = 150      MFIB RRP RF Client
clientID = 402      group_id = 1        clientSeq = 161      TPM RF client
clientID = 520      group_id = 1        clientSeq = 162      RFS RF
clientID = 210      group_id = 1        clientSeq = 163      Auth Mgr
clientID = 10101    group_id = 1        clientSeq = 164      NGMOD HMS RF client
clientID = 5        group_id = 1        clientSeq = 165      Config Sync RF client
clientID = 10007    group_id = 1        clientSeq = 170      NGWC FEC Rf client
clientID = 10009    group_id = 1        clientSeq = 173      NGWC POWERNET Rf client
clientID = 10100    group_id = 1        clientSeq = 174      NGMOD XCVR RF client
clientID = 502      group_id = 1        clientSeq = 187      LAN-Switch Port Manager
clientID = 530      group_id = 1        clientSeq = 189      Access Tunnel
clientID = 519      group_id = 1        clientSeq = 190      Mac address Table Manager
clientID = 209      group_id = 1        clientSeq = 209      L2FIB
clientID = 207      group_id = 1        clientSeq = 215      CFM RF
clientID = 208      group_id = 1        clientSeq = 218      LLDP
clientID = 226      group_id = 1        clientSeq = 219      LACP
```

The following is sample output from the **show redundancy counters** command:

```
Device# show redundancy counters

Redundancy Facility OMs
              comm link up = 0
            comm link down = 0

          invalid client tx = 0
          null tx by client = 0
                tx failures = 0
      tx msg length invalid = 0

      client not rxing msgs = 0
 rx peer msg routing errors = 0
           null peer msg rx = 0
         errored peer msg rx = 0

                 buffers tx = 7250
     tx buffers unavailable = 0
                 buffers rx = 6786
      buffer release errors = 0

 duplicate client registers = 0
  failed to register client = 0
       Invalid client syncs = 0
```

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

```
Device# show redundancy states

       my state = 13 -ACTIVE
     peer state = 1  -DISABLED
           Mode = Simplex
           Unit = Primary
        Unit ID = 3

Redundancy Mode (Operational) = Non-redundant
Redundancy Mode (Configured)  = sso
Redundancy State              = Non Redundant
     Maintenance Mode = Disabled
    Manual Swact = disabled (system is simplex (no peer unit))
 Communications = Down      Reason: Simplex mode
```

```
  client count = 103
client_notification_TMR = 30000 milliseconds
          RF debug mask = 0x0
```

Additional References for Cisco Nonstop Forwarding with Stateful Switchover

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the *High Availability* section of the *Command Reference (Catalyst 9400 Series Switches)*

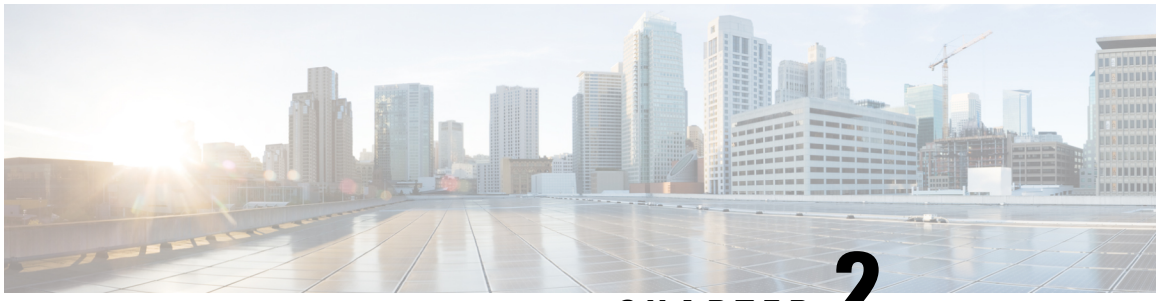
Feature History for Cisco Nonstop Forwarding with Stateful Switchover

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 16.6.2	Cisco Nonstop Forwarding with Stateful Switchover	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.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com.



CHAPTER 2

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 13
- Information About Graceful Insertion and Removal, on page 13
- How to Configure Graceful Insertion and Removal, on page 16
- Monitoring Graceful Insertion and Removal, on page 18
- Configuration Examples for Graceful Removal and Insertion, on page 18
- Additional References for Graceful Insertion and Removal , on page 20
- Feature History for Graceful Insertion and Removal, on page 20

Restrictions for Graceful Insertion and Removal

- The BGP instance in the maintenance template should match the BGP instance on the device. Also, do not configure more than one BGP instances under the maintenance template.
- 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 **shut down 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 onthe 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-namesnapshot-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***calss** 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_IGP, 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

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**maintenance-template** *template_name* **Example:** `Device(config)# maintenance-template gir1`	Creates a template with the specified name. For example, see Examples: Creating customer profile.

Step 4	**router** *routing_protocol instance_id* | **shutdown l2**
	Example:
	`Device(config-maintenance-templ)# router isis 1`
	`Device(config-maintenance-templ)# shutdown l2`
	`Device(config-maintenance-templ)# router bgp AS-number`
		Creates instances that should be isolated under this template.
		• **router:** Configures routing protocols and associated instance id.
		• **shutdown l2:** Shuts down layer 2 interfaces.

Configuring System Mode Maintenance

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.

	Command or Action	Purpose
Step 3	**system mode maintenance** **Example:** `Device(config)# system mode maintenance`	Enters system mode maintenance configuration mode. Different sub commands to create maintenance mode parameters are configured in this mode.
Step 4	**timeout** *timeout-value* | **template** *template-name* | **failsafe** *failsafe-timeout-value* | **on-reload reset-reason maintenance**	Configures maintenance mode parameters. • **timeout:** Configures maintenance mode timeout period in minutes, after which the system automatically returns to normal mode. The default timeout value is never. • **template:** Configures maintenance mode using the specified template. • **failsafe:**Configures client-ack timeout value. 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. • **on-reload reset-reason maintenance:**Configures the system such that when the system is reloaded it enters the maintenance mode. If it is not configured the system enters the normal mode when it is reloaded.

Starting and Stopping Maintenance Mode

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**start maintenance** **Example:** `Device# start maintenance`	Puts the system into maintenance mode.
Step 3	**stop maintenance** **Example:** `Device# stop maintenance`	Puts the system back into normal mode.

Monitoring Graceful Insertion and Removal

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

Table 1: Privileged EXEC Commands

Command	Purpose
show system mode [maintenance [clients | template *template-name*]]	Displays information about system mode.
show system snapshots [dump *<snapshot-file-name>]*	Displays all the snapshots present on the device.
show system snapshots [dump *<snapshot-file-name>]***xml**	Displays all the snapshots present on the device in XML format.
show system snapshots compare *snapshot-name1 snapshot-name2*	Displays differences between snapshots taken before entering maintenance mode and after exiting from the maintenance mode.

Table 2: Global Configuration Commands for Troubleshooting

Command	Purpose
debug system mode maintenance	Displays information to help troubleshoot the GIR feature.

Configuration Examples for Graceful Removal and Insertion

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

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

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)# system mode maintenance
Device(config-maintenance)# timeout 20
Device(config-maintenance)# failsafe 30
Device(config-maintenance)# on-reload reset-reason maintenance
Device(config-maintenance)# template t1
Device(config-maintenance)# exit
```

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:
```

```
     router isis 1
     router isis 2

Device# show system mode maintenance template t1
     System Mode: Normal
     Maintenance Template t1 details:
     router isis 1
```

Additional References for Graceful Insertion and Removal

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the *High Availability* section of the *Command Reference (Catalyst 9400 Series Switches)*

Feature History for Graceful Insertion and Removal

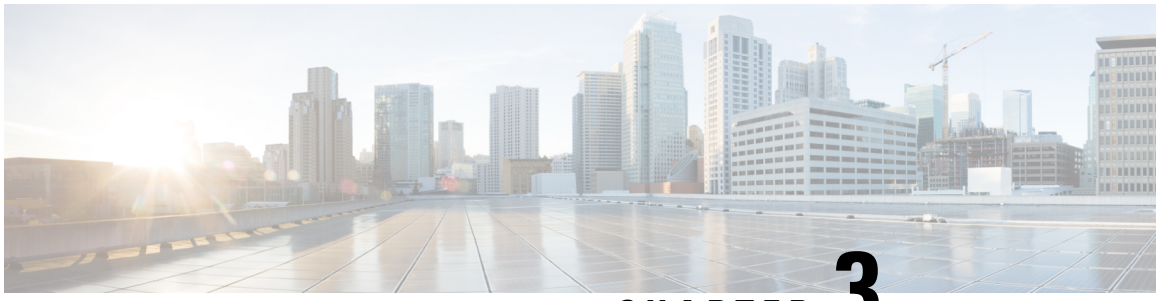
This table provides release and related information for features explained in this module.

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 Fuji 16.8.1a	Graceful Insertion and Removal	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.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.1	Graceful Insertion and Removal (GIR) enhancements: snapshot templates	The following enhancements were introduced: • 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 Layer 2 protocol support for GIR Hot Standby Router Protocol (HSRP)	GIR is now supported for the HSRP protocol.
	GIR Layer 2 protocol support for GIR Virtual Router Redundancy Protocol (VRRP)	GIR is now supported for VRRPv3 protocol.
Cisco IOS XE Gibraltar 16.10.1	Graceful Insertion and Removal (GIR) Support for BGP	GIR is now supported for the BGP protocol.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com.



CHAPTER 3

Configuring Cisco StackWise Virtual

- Prerequisites for Cisco StackWise Virtual, on page 23
- Restrictions for Cisco StackWise Virtual, on page 24
- Information About Cisco StackWise Virtual, on page 25
- How to Configure Cisco StackWise Virtual, on page 38
- Configuration Examples for StackWise Virtual, on page 47
- Verifying Cisco StackWise Virtual Configuration, on page 49
- Additional References for StackWise Virtual, on page 50
- Feature History for Cisco StackWise Virtual, on page 50

Prerequisites for Cisco StackWise Virtual

- All the switches in a Cisco StackWise Virtual solution must be of the same switch model.
- The supervisor modules in a Cisco StackWise Virtual solution must be of the same model.
- The supervisor module in each switch should be inserted in symmetrical slots. For example, in a Cisco Catalyst 9407R Switch, if you have installed the supervisor module in slot 3, ensure that the second switch also has the supervisor module installed in slot 3.

  For chassis slot restrictions, refer to the Cisco Catalyst 9400 Series Supervisor Module Installation Note.
- All the switches in a Cisco StackWise Virtual solution must be running the same license level.
- All the switches in a Cisco StackWise Virtual solution must be running the same software version.
- All the switches in a Cisco StackWise Virtual solution must be running the same SDM template.
- All the ports used for configuring a StackWise Virtual Link (SVL) must share the same speed. For example, you cannot configure a 10G or a 40G port to form an SVL, simultaneously.
- When configuring SVLs and dual-active detection (DAD) links on line cards, we recommend that you perform the following:
  - Enable autoLC shutdown on line cards. Auto line card shutdown feature allows you to configure line card power priority to enable the hardware to automatically shutdown least priority line cards in power constraint mode.

- Configure higher priority for line cards that have SVLs and DAD links configured. This ensures that in situations with insufficient power, the linecards with SVLs and DAD links will be reloaded last.

You can configure autoLC shutdown and line card power priority by using the **power supply autoLC** [**priority** *physical-slot-number*] [**shutdown**] command in global configuration mode.

On a switch stack, use the **power supply switch** *switch-number* **autoLC** [**priority** *physical-slot-number*] [**shutdown**] command.

Restrictions for Cisco StackWise Virtual

- Cisco StackWise Virtual is supported on Cisco Catalyst 9400 Series Supervisor 1 Module (C9400-SUP-1) and Cisco Catalyst 9400 Series Supervisor 1XL Module (C9400-SUP-1XL). The feature requires a special, additional, C9400-SUP-UPG-LIC= license with Cisco Catalyst 9400 Series Supervisor 1 Module (C9400-SUP-1).
- Cisco StackWise Virtual configuration commands will be recognised only on a switch running Network Advantage license. The configuration commands will not be recognised on a Network Essentials license.
- Cisco StackWise Virtual can be configured only on one supervisor module per chassis. You must not install two supervisor modules in each chassis used in the Cisco StackWise Virtual solution.
- When deploying Cisco StackWise Virtual, ensure that VLAN ID 4094 is not used anywhere on the network. All inter-chassis system control communication between stack members is carried over the reserved VLAN ID 4094 from the global range.
- Dual-active detection (DAD) and StackWise Virtual (SVL) link configuration must be performed manually and the device should be rebooted for the configuration changes to take effect.
- Only Cisco Transceiver Modules are supported.
- The interface VLAN MAC address that is assigned by default, can be overridden using the **mac-address** command. If this command is configured on a single SVI or router port that requires Layer 3 injected packets, all other SVIs or routed ports on the device also must be configured with the same first four most significant bytes (4MSB) of the MAC address. For example, if you set the MAC address of any SVI to xxxx.yyyy.zzzz, set the MAC address of all other SVIs to start with xxxx.yyyy. If Layer 3 injected packets are not used, this restriction does not apply.

Note This applies to all Layer 3 ports, SVIs, and routed ports. This does not apply to GigabitEthernet0/0 port.

- Cisco Catalyst 9400 Series Switches do not support pre-provisioning. This affects how the standby or active switch configuration is stored in the running and startup configuration. If an active switch is powered off, during the active switch changeover, the existing active switch will be removed from the running configuration of the new switch. Executing the write command during the changeover will remove the active switch configuration from the startup configuration also.

  If a standby switch is powered off, the standby switch configuration will be removed from the running configuration. Executing the write command will remove the standby configuration from the startup configuration also. The standby configuration will be re-learnt when the standby switch re-joins the stack.

- Broadcast, Unknown Unicast and Multicast (BUM) Traffic Optimization is not applicable to VLANs with standalone or physical ports.

Information About Cisco StackWise Virtual

Cisco StackWise Virtual on Cisco Catalyst 9400 Series Switches

This section describes the Cisco StackWise Virtual features specific to Cisco Catalyst 9400 Series Switches.

- Cisco StackWise Virtual is supported on Cisco Catalyst 9404R, Cisco Catalyst 9407R, and Catalyst 9410R switches.
- You can configure SVLs and DAD links on Cisco Catalyst 9400 Series supervisor modules and select Ethernet switching modules (line cards). SVL connections are established through 10G, 40G or 25G (available only on C9400-SUP-1XL-Y) uplink ports on the supervisor modules and 10G downlink ports on the line cards. For information about supported supervisor modules and line cards, see the following table.

  The following table provides a matrix of StackWise Virtual communication mechanism on each module:

Table 3: StackWise Virtual Feature Matrix for Supervisor Modules

Product ID	StackWise Virtual Link	Dual-Active Detection Link
Supervisor Modules		
C9400-SUP-1	Supported	Supported
C9400-SUP-1XL	Supported	Supported
C9400-SUP-1XL-Y	Supported	Supported

Table 4: StackWise Virtual Feature Matrix for Line Cards

Product ID	StackWise Virtual Link	Dual-Active Detection Link
Gigabit Ethernet Switching Modules		
C9400-LC-24S	Not Supported	Supported
C9400-LC-48P	Not Supported	Supported
C9400-LC-48S	Not Supported	Supported
C9400-LC-48T	Not Supported	Supported
C9400-LC-48U	Not Supported	Supported
Ten Gigabit Ethernet Switching Modules		
C9400-LC-24XS	Supported	Supported

Product ID	StackWise Virtual Link	Dual-Active Detection Link
Multigigabit Ethernet Switching Modules		
C9400-LC-48UX	Supported Multigigabit (mGig) ports 25 to 48)	Supported

- 25G links can be established only through uplink ports 1 and 5 of C9400-SUP-1XL-Y. If you enable SVLs or DAD links on the 25G ports, the corresponding 10G and 40G ports on the modules are disabled. For example, if TwentyFiveGigE1/2/0/1 is configured as the SVL port, the ports - TenGigabitEthernet1/2/0/1 to TenGigabitEthernet1/2/0/4 and FortyGigabitEthernet1/2/0/9 are disabled. Similarly for a 40G port, if port FortyGigabitEthernet1/2/0/9 is configured as the SVL port, ports TenGigabitEthernet1/2/0/1 to TenGigabitEthernet1/2/0/4 and TwentyFiveGigE1/2/0/1 are disabled.

  For more information about configuring uplink ports on supervisor modules, see "Uplink Ports" section in "Configuring Interface Characteristics" chapter of the *Interface and Hardware Components Configuration Guide for Catalyst 9400 Switches*.

- You can configure up to 8 SVLs in a Cisco StackWise Virtual solution using Cisco Catalyst 9400 Series Switches.
- SVLs can have up to 80GE (8x10GE or 2x40GE) or 50GE (2x25GE) combined bandwidth per chassis.
- After configuring StackWise Virtual in a standalone chassis and rebooting the switches to form a stack, the interface naming convention changes from the default 3-tuple (slot/bay/port) to 4-tuple (chassis/slot/bay/port) which includes the chassis identifier as part of the interface name. For example, Gi2/0/1 will change to Gi1/2/0/1 where the first number denotes the chassis number.

  The following points explain changes to SNMP with the introduction of 4-tuple interface naming convention:

  - The physical index for chassis 1 and 2 is 2 and 500 respectively.
  - The physical index for the slots in chassis 1 is in the range of 1000-10000 and slots in chassis 2 is in the range of 11000-20000.
  - All MIB Object Identifiers (OID) that require a slot number for query will use a physical slot index in the flat number space {1, 2, 3...20}, where slots 1-10 indicate chassis 1 and slots 11-20 indicate chassis 2.
  - Use **show snmp slot-mapping** command to display the chassis/slot mapping.

Supervisor Modules/Line Cards - Supported Combinations

Note An SVL can be formed only between the supported combinations listed below.

- Supported Combination 1: Any two Cisco Catalyst 9400 Series supervisor modules of the same model.
- Supported Combination 2: 10G uplink ports on any Cisco Catalyst 9400 Series supervisor module + C9400-LC-24XS

- Supported Combination 3: C9400-LC-24XS + C9400-LC-24XS
- Supported Combination 4: C9400-LC-48UX + C9400-LC-48UX

Note For information about supported supervisor modules and line cards, see Table 3: StackWise Virtual Feature Matrix for Supervisor Modules, on page 25.

Overview of Cisco StackWise Virtual

Cisco StackWise Virtual is a network system virtualization technology that pairs two directly connected switches into one virtual switch. The switches in a Cisco StackWise Virtual solution increase operational efficiency by using single control and management plane, scale system bandwidth with distributed forwarding plane, and help in building resilient networks using the recommended network design. Cisco StackWise Virtual allows two directly connected physical switches to operate as a single logical virtual switch using an Ethernet connection.

Cisco StackWise Virtual Topology

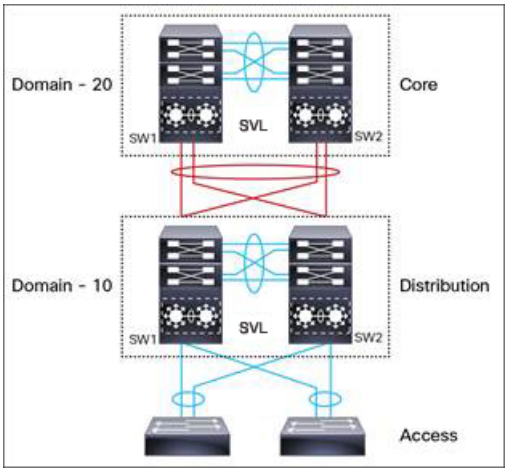
A typical network design consists of core, distribution, and access layers. The default mode of a switch is standalone. When two redundant switches are deployed in the distribution layer, the following network challenges arise:

- If VLAN IDs are reused between access layers then, it will introduce a spanning tree loop that will impact the overall performance of the network.
- Spanning tree protocols and configuration are required to protect Layer 2 network against spanning tree protocol loop, and root and bridge protocol data unit management.
- Additional protocols such as first hop redundancy protocol are required to virtualize the IP gateway function. This should align with STP root priorities for each VLAN.
- The Protocol independent multicast designated router (PIM DR) configuration should be fine-tuned to selectively build a multicast forwarding topology on a VLAN.
- The standalone distribution layer system provides protocol-driven remote failure and detection, which results in slower convergence time. Fine-tune FHRP and PIM timers for rapid fault detection and recovery process.

We recommend Cisco StackWise Virtual model for aggregation layers and collapsed aggregation and core layers. The stack can be formed over a 25G, 40G or 10G links to ensure that the distribution or the aggregation switches can be deployed over a large distance.

Note that STP keeps one of the ports connected to the distribution switches blocked on the access switches. As a result of this, an active link failure causes STP convergence and the network suffers from traffic loss, flooding, and a possible transient loop in the network. On the other hand, if the switches are logically merged into one switch, all the access switches might form an EtherChannel bundle with distribution switches, and a link failure within an EtherChannel would not have any impact as long as at least one member within the EtherChannel is active.

Figure 1: Typical Network Design using Cisco StackWise Virtual



Etherchannel in StackWise Virtual is capable of implementing Multi-chassis EtherChannel (MEC) across the stack members. When access layer and aggregation layer are collapsed into a single StackWise Virtual system, MEC across the different access layer domain members and across distribution and access layer switches will not be supported. MEC is designed to forward the traffic over the local link irrespective of the hash result.

Since the control plane, management plane, and data plane are integrated, the system behaves as a single switch.

The virtualization of multiple physical switches into a single logical switch is from a control and management plane perspective only. Because of the control plane being common, it may look like a single logical entity to peer switches. The data plane of the switches is distributed. Each switch is capable of forwarding over its local interfaces without involving other members. However, when a packet coming into a switch has to be forwarded over a different member's port, the forwarding context of the packet is carried over to the destination switch after ingress processing is performed in the ingress switch. Egress processing is done only in the egress switch. This provides a uniform data plane behavior to the entire switch irrespective whether of the destination port is in a local switch or in a remote switch. However, the common control plane ensures that all the switches have equivalent data plane entry for each forwarding entity.

An election mechanism elects one of the switches to be Cisco StackWise Virtual active and the other switch to be Cisco StackWise Virtual standby in terms of Control Plane functions. The active switch is responsible for all the management, bridging and routing protocols, and software data path. The standby switch is in hot standby state ready to take over the role of active, if the active switch fails over.

The following are the components of the Cisco StackWise Virtual solution:

- Stack members
- SVL: 25G, 40G or 10G Ethernet connections. SVL is established using the 25G, 40G or 10G interfaces depending on the switch models. However, a combination of two different speeds is not supported.

SVL is the link that connects the switches over Ethernet. Typically, Cisco StackWise Virtual consists of multiple 25G, 40G or 10G physical links. It carries all the control and data traffic between the switching units. You can configure SVL on a supported port. When a switch is powered up and the hardware is initialized, it looks for a configured SVL before the initialization of the control plane.

The Link Management Protocol (LMP) is activated on each link of the SVL as soon as the links are established. LMP ensure the integrity of the links and monitors and maintains the health of the links. The redundancy role of each switch is resolved by the StackWise Discovery Protocol (SDP). It ensures that the hardware and

software versions are compatible to form the SVL and determines which switch becomes active or standby from a control plane perspective.

Cisco StackWise Virtual Header (SVH) is 64-byte frame header that is prepended over all control, data, and management plane traffic that traverse over each SVL between the two stack members of the Cisco StackWise Virtual domain. The SVH-encapsulated traffic operates at OSI Layer 2 and can be recognized and processed only by Cisco StackWise Virtual-enabled switches. SVL interfaces are non-bridgeable and non-routeable, and allows non-routeable traffic over L2 or L3 network.

Cisco StackWise Virtual Redundancy

Cisco StackWise Virtual operates stateful switchover (SSO) between the active and standby switches. The following are the ways in which Cisco StackWise Virtual's redundancy model differs from that of the standalone mode:

- The Cisco StackWise Virtual active and standby switches are hosted in separate switches and use a StackWise Virtual link to exchange information.
- The active switch controls both the switches of Cisco StackWise Virtual. The active switch runs the Layer 2 and Layer 3 control protocols and manages the switching modules of both the switches.
- The Cisco StackWise Virtual active and standby switches perform data traffic forwarding.

Note If the Cisco StackWise Virtual active switch fails, the standby switch initiates a switchover and assumes the Cisco StackWise Virtual active switch role.

SSO Redundancy

A StackWise Virtual system operates with SSO redundancy if it meets the following requirements:

- Both the switches must be running the same software version, unless they are in the process of software upgrade.
- SVL-related configuration in the two switches must match.
- License type must be same on both the switch models.
- Both the switch models must be in the same StackWise Virtual domain.

With SSO redundancy, the StackWise Virtual standby switch is always ready to assume control if a fault occurs on the StackWise Virtual active switch. Configuration, forwarding, and state information are synchronized from the StackWise Virtual active switch to the redundant switch at startup, and whenever changes to the StackWise Virtual active switch configuration occur. If a switchover occurs, traffic disruption is minimized.

If StackWise Virtual does not meet the requirements for SSO redundancy, it will be incapable of establishing a relationship with the peer switch. StackWise Virtual runs stateful switchover (SSO) between the StackWise Virtual active and standby switches. The StackWise Virtual determines the role of each switch during initialization.

The CPU in the StackWise Virtual standby switch runs in hot standby state. StackWise Virtual uses SVL to synchronize configuration data from the StackWise Virtual active switch to the StackWise Virtual standby

switch. Also, protocols and features that support high availability synchronize their events and state information to the StackWise Virtual standby switch.

Nonstop Forwarding

While implementing Nonstop Forwarding (NSF) technology in systems using SSO redundancy mode, network disruptions are minimized for campus users and applications. High availability is provided even when the control-plane processing stack-member switch is reset. During a failure of the underlying Layer 3, NSF-capable protocols perform graceful network topology resynchronization. The preset forwarding information on the redundant stack-member switch remains intact; this switch continues to forward the data in the network. This service availability significantly lowers the mean time to repair (MTTR) and increases the mean time between failure (MTBF) to achieve a high level of network availability.

Multichassis EtherChannels

Multichassis EtherChannel (MEC) is an EtherChannel bundled with physical ports having common characteristics such as speed and duplex, that are distributed across each Cisco StackWise Virtual system. A Cisco StackWise Virtual MEC can connect to any network element that supports EtherChannel (such as a host, server, router, or switch). Cisco StackWise Virtual supports up to 252 MECs deployed in Layer 2 or Layer 3 modes. EtherChannel 127 and 128 is reserved for SVL connections. Hence, the maximum available MEC count is 250.

In a Cisco StackWise Virtual system, an MEC is an EtherChannel with additional capability. A multichassis EtherChannel link reduces the amount of traffic that requires transmission across the SVL by populating the index port only with the ports local to the physical switch. This allows the switch to give precedence to the local ports of the multichassis EtherChannel link over those on the remote switch.

Each MEC can optionally be configured to support either Cisco PAgP, IEEE LACP, or Static ON mode. We recommend that you implement EtherChannel using Cisco PAgP or LACP with a compatible neighbor. If a remotely connected neighbor such as Cisco Wireless LAN Controller (WLC) does not support this link-bundling protocol, then a Static ON mode can be deployed. These protocols run only on the Cisco StackWise Virtual active switch.

An MEC can support up to eight physical links that can be distributed in any proportion between the Cisco StackWise Virtual active switch and the Cisco StackWise Virtual standby switch. We recommend that you distribute the MEC ports across both switches evenly.

MEC Minimum Latency Load Balancing

The StackWise Virtual environment is designed such that data forwarding always remains within the switch. The Virtual Stack always tries to forward traffic on the locally available links. This is true for both Layer 2 and Layer3 links. The primary motivation for local forwarding is to avoid unnecessarily sending data traffic over the SVL and thus reduce the latency (extra hop over the SVL) and congestion.The bidirectional traffic is load-shared between the two StackWise Virtual members. However, for each StackWise Virtual member, ingress and egress traffic forwarding is based on locally-attached links that are part of MEC. This local forwarding is a key concept in understanding convergence and fault conditions in a StackWise Virtual enabled campus network.

The active and standby switches support local forwarding that will individually perform the desired lookups and forward the traffic on local links to uplink neighbors. If the destination is a remote switch in the StackWise Virtual domain, ingress processing is performed on the ingress switch and then traffic is forwarded over the SVL to the egress switch where only egress processing is performed.

MEC Failure Scenarios

The following sections describe issues that may arise and the resulting impact:

Single MEC Link Failure

If a link within a MEC fails (and other links in the MEC are still operational), the MEC redistributes the load among the operational links, as in a regular port.

All MEC Links to the Cisco StackWise Virtual Active Switch Fail

If all the links to the Cisco StackWise Virtual active switch fail, a MEC becomes a regular EtherChannel with operational links to the Cisco StackWise Virtual standby switch.

Data traffic that terminates on the Cisco StackWise Virtual active switch reaches the MEC by crossing the SVL to the Cisco StackWise Virtual standby switch. Control protocols continue to run in the Cisco StackWise Virtual active switch. Protocol messages reach the MEC by crossing the SVL.

All MEC Links Fail

If all the links in an MEC fail, the logical interface for the EtherChannel is set to Unavailable. Layer 2 control protocols perform the same corrective action as for a link-down event on a regular EtherChannel.

On adjacent switches, routing protocols and the Spanning Tree Protocol (STP) perform the same corrective action as for a regular EtherChannel.

Cisco StackWise Virtual Standby Switch Failure

If the Cisco StackWise Virtual standby switch fails, a MEC becomes a regular EtherChannel with operational links on the Cisco StackWise Virtual active switch. Connected peer switches detect the link failures, and adjust their load-balancing algorithms to use only the links to the StackWise Virtual active switch.

Cisco StackWise Virtual Active Switch Failure

Cisco StackWise Virtual active switch failure results in a stateful switchover (SSO). After the switchover, a MEC is operational on the new Cisco StackWise Virtual active switch. Connected peer switches detect the link failures (to the failed switch), and adjust their load-balancing algorithms to use only the links to the new Cisco StackWise Virtual active switch.

Cisco StackWise Virtual Packet Handling

In Cisco StackWise Virtual, the Cisco StackWise Virtual active switch runs the Layer 2 and Layer 3 protocols and features and manages the ports on both the switches.

Cisco StackWise Virtual uses StackWise Virtual link to communicate system and protocol information between the peer switches and to carry data traffic between the two switches.

The following sections describe packet handling in Cisco StackWise Virtual.

Traffic on StackWise Virtual link

SVL carries data traffic and in-band control traffic between two switches. All the frames that are forwarded over the SVL are encapsulated with a special StackWise Virtual Header (SVH). The SVH adds an overhead

of 64 bytes for control and data traffic, which provides information for Cisco StackWise Virtual to forward the packet on the peer switch.

An SVL transports control messages between two switches. Messages include protocol messages that are processed by the Cisco StackWise Virtual active switch, but received or transmitted by interfaces on the Cisco StackWise Virtual standby switch. Control traffic also includes module programming between the Cisco StackWise Virtual active switch and the switching modules on the Cisco StackWise Virtual standby switch.

Cisco StackWise Virtual transmits data traffic over an SVL under the following circumstances:

- Layer 2 traffic flooded over a VLAN (even for dual-homed links).
- Packets processed by software on the Cisco StackWise Virtual active switch where the ingress interface is on the Cisco StackWise Virtual standby switch.
- The packet destination is on the peer switch, as described in the following examples:
  - Traffic within a VLAN where the known destination interface is on the peer switch.
  - Traffic that is replicated for a multicast group and the multicast receivers are on the peer switch.
  - The known unicast destination MAC address is on the peer switch.
  - The packet is a MAC notification frame destined for a port on the peer switch.

An SVL also transports system data, such as NetFlow export data and SNMP data, from the Cisco StackWise Virtual standby switch to the Cisco StackWise Virtual active switch.

Traffic on the SVL is load balanced with the same global hashing algorithms available for EtherChannels (the default algorithm is source-destination IP).

Layer 2 Protocols

The Cisco StackWise Virtual active switch runs the Layer 2 protocols (such as STP and VTP) for the switching modules on both the switches. Protocol messages that are received on the standby switch ports must traverse SVLs to reach the active switch where they are processed. Similarly, protocol messages that are transmitted from the standby switch ports originate on the active switch, and traverse the SVLs to reach the standby ports.

All the Layer 2 protocols in Cisco StackWise Virtual work similarly in standalone mode. The following sections describe the difference in behavior for some protocols in Cisco StackWise Virtual.

Spanning Tree Protocol

The Cisco StackWise Virtual active switch runs the STP. The Cisco StackWise Virtual standby switch redirects the STP BPDUs across an SVL to the StackWise Virtual active switch.

The STP bridge ID is commonly derived from the switch MAC address. To ensure that the bridge ID does not change after a switchover, Cisco StackWise Virtual continues to use the original switch MAC address for the STP Bridge ID.

EtherChannel Control Protocols

Link Aggregation Control Protocol (LACP) and Port Aggregation Protocol (PAgP) packets contain a device identifier. Cisco StackWise Virtual defines a common device identifier for both the switches. Use either PAgP or LACP on Multi EtherChannels instead of mode ON, even if all the three modes are supported.

Note A new PAgP enhancement has been defined for assisting with dual-active scenario detection.

Switched Port Analyzer

Switched Port Analyzer (SPAN) on SVL and fast hello DAD link ports is not supported. These ports can be neither a SPAN source, nor a SPAN destination. Cisco StackWise Virtual supports all the SPAN features for non-SVL interfaces. The number of SPAN sessions that are available on Cisco StackWise Virtual matches that on a single switch running in standalone mode.

Private VLANs

Private VLANs on StackWise Virtual work the same way as in standalone mode. The only exception is that the native VLAN on isolated trunk ports must be configured explicitly.

Apart from STP, EtherChannel Control Protocols, SPAN, and private VLANs, the Dynamic Trunking Protocol (DTP), Cisco Discovery Protocol (CDP), VLAN Trunk Protocol (VTP), and Unidirectional Link Detection Protocol (UDLD) are the additional Layer 2 control-plane protocols that run over the SVL connections.

Broadcast, Unknown Unicast and Multicast

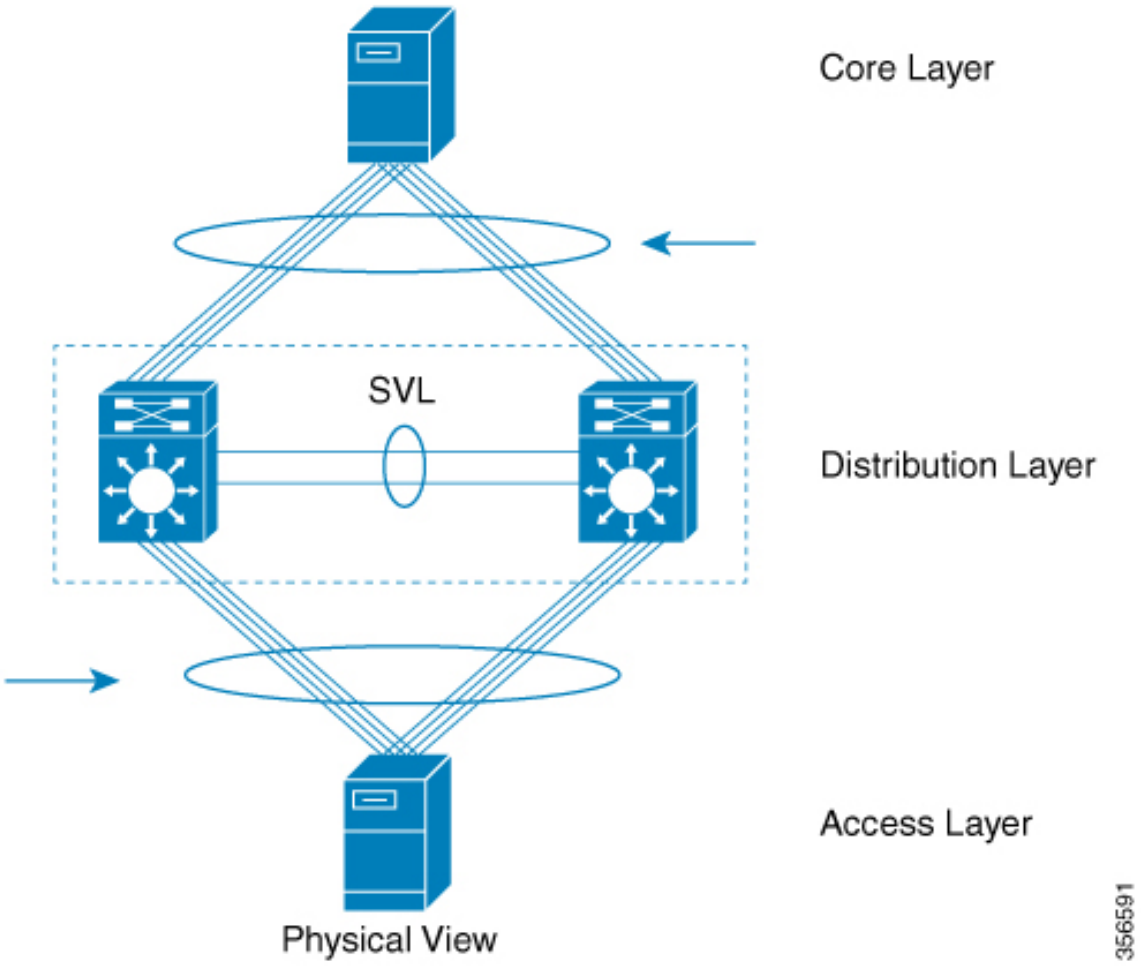
Cisco StackWise Virtual supports local switching for Broadcast, Unknown unicast and Multicast (BUM) traffic. In uncommon deployment scenarios, BUM traffic traverses through the StackWise Virtual Links. This section explains how BUM traffic is handled in a Cisco StackWise Virtual setup and in local switching.

When a VLAN is created, StackWise Virtual ports are added to the VLAN flood list. The ingress BUM traffic on active or standby switch traverses through the StackWise Virtual link to the other switch instead of a port in the VLAN. This traffic floods the StackWise Virtual links which impacts the system and network performance.

To address this, StackWise Virtual BUM optimization feature is introduced.

A general deployment guideline for Cisco StackWise Virtual is to distribute MEC ports evenly at the uplink and downlink as shown in the figure. In this topology, BUM traffic prefers the local link on MEC to send the traffic out instead of the StackWise Virtual link. In a scenario where there is a standalone port on a switch or members of EtherChannel on active or standby switch are down, BUM traffic traverses the StackWise Virtual link. When StackWise Virtual BUM optimization is enabled on VLAN, StackWise Virtual port is not added to the VLAN flood list. This design ensures BUM traffic does not traverse StackWise Virtual link only when MEC port channels are part of the VLAN. No optimization is done for VLANs with standalone or physical ports.

Figure 2: Recommended Topology for Cisco StackWise Virtual



Layer 3 Protocols

The Cisco StackWise Virtual active switch runs the Layer 3 protocols and features for the StackWise Virtual. All the Layer 3 protocol packets are sent to and processed by the Cisco StackWise Virtual active switch. Both the member switches perform hardware forwarding for ingress traffic on their interfaces. When software forwarding is required, packets are sent to the Cisco StackWise Virtual active switch for processing.

The same router MAC address assigned by the Cisco StackWise Virtual active switch is used for all the Layer 3 interfaces on both the Cisco StackWise Virtual member switches. After a switchover, the original router MAC address is still used. The router MAC address is chosen based on chassis-mac and is preserved after switchover by default.

The following sections describe the Layer 3 protocols for Cisco StackWise Virtual.

IPv4 Unicast

The CPU on the Cisco StackWise Virtual active switch runs the IPv4 routing protocols and performs any required software forwarding. All the routing protocol packets received on the Cisco StackWise Virtual standby

switch are redirected to the Cisco StackWise Virtual active switch across the SVL. The Cisco StackWise Virtual active switch generates all the routing protocol packets to be sent out over ports on either of the Cisco StackWise Virtual member switches.

Hardware forwarding is distributed across both members on Cisco StackWise Virtual. The CPU on the Cisco StackWise Virtual active switch sends Forwarding Information Base (FIB) updates to the Cisco StackWise Virtual standby switch, which in turn installs all the routes and adjacencies into hardware.

Packets intended for a local adjacency (reachable by local ports) are forwarded locally on the ingress switch. Packets intended for a remote adjacency (reachable by remote ports) must traverse the SVL.

The CPU on the Cisco StackWise Virtual active switch performs all software forwarding and feature processing (such as fragmentation and Time to Live exceed functions). If a switchover occurs, software forwarding is disrupted until the new Cisco StackWise Virtual active switch obtains the latest Cisco Express Forwarding and other forwarding information.

In virtual switch mode, the requirements to support non-stop forwarding (NSF) match those in the standalone redundant mode of operation.

From a routing peer perspective, Multi-Chassis EtherChannels (MEC) remain operational during a switchover, that is, only the links to the failed switch are down, but the routing adjacencies remain valid.

Cisco StackWise Virtual achieves Layer 3 load balancing over all the paths in the Forwarding Information Base entries, be it local or remote.

IPv6

Cisco StackWise Virtual supports IPv6 unicast and multicast because it is present in the standalone system.

IPv4 Multicast

The IPv4 multicast protocols run on the Cisco StackWise Virtual active switch. Internet Group Management Protocol (IGMP) and Protocol Independent Multicast (PIM) protocol packets received on the Cisco StackWise Virtual standby switch are transmitted across an SVL to the StackWise Virtual active switch. The latter generates IGMP and PIM protocol packets to be sent over ports on either of the Cisco StackWise Virtual members.

The Cisco StackWise Virtual active switch synchronizes the Multicast Forwarding Information Base (MFIB) state to the Cisco StackWise Virtual standby switch. On both the member switches, all the multicast routes are loaded in the hardware, with replica expansion table (RET) entries programmed for only local, outgoing interfaces. Both the member switches are capable of performing hardware forwarding.

Note To avoid multicast route changes as a result of a switchover, we recommend that all the links carrying multicast traffic be configured as MEC rather than Equal Cost Multipath (ECMP).

For packets traversing an SVL, all Layer 3 multicast replications occur on the egress switch. If there are multiple receivers on the egress switch, only one packet is replicated and forwarded over the SVL, and then replicated to all the local egress ports.

Software Features

Software features run only on the Cisco StackWise Virtual active switch. Incoming packets to the Cisco StackWise Virtual standby switch that require software processing are sent across an SVL to the Cisco StackWise Virtual active switch.

Dual-Active Detection

If the standby switch detects a complete loss of the SVL, it assumes the active switch has failed and will take over as the active switch. However, if the original Cisco StackWise Virtual active switch is still operational, both the switches will now be Cisco StackWise Virtual active switches. This situation is called a dual-active scenario. This scenario can have adverse effects on network stability because both the switches use the same IP addresses, SSH keys, and STP bridge IDs. Cisco StackWise Virtual detects a dual-active scenario and takes recovery action. DAD link is the dedicated link used to mitigate this.

If the last available SVL fails, the Cisco StackWise Virtual standby switch cannot determine the state of the Cisco StackWise Virtual active switch. To ensure network uptime without delay, the Cisco StackWise Virtual standby switch then assumes the Cisco StackWise Virtual active role. The original Cisco StackWise Virtual active switch enters recovery mode and brings down all its interfaces, except the SVL and the management interfaces.

Dual-Active-Detection Link with Fast Hello

To use the dual-active fast hello packet detection method, you must provision a direct ethernet connection between the two Cisco StackWise Virtual switches. You can dedicate up to four links for this purpose.

The two switches periodically exchange special dual-active hello messages containing information about the switch state. If all SVLs fail and a dual-active scenario occurs, each switch recognizes that there is a dual-active scenario from the peer's messages.

This initiates recovery actions as described in the Recovery Actions, on page 37 section. If a switch does not receive an expected dual-active fast hello message from the peer before the timer expires, the switch assumes that the link is no longer capable of dual-active detection.

Note Do not use the same port for StackWise Virtual Link and dual-active detection link.

Dual-Active Detection with enhanced PAgP

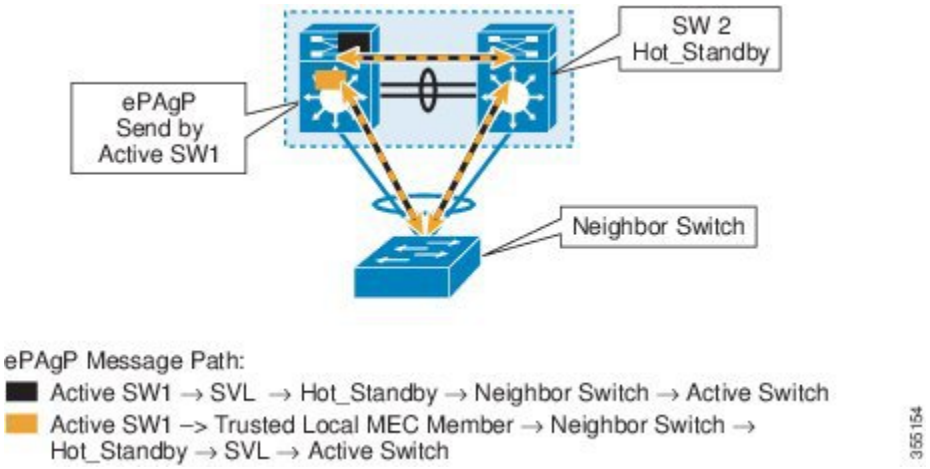
Port aggregation protocol (PAgP) is a Cisco proprietary protocol used for managing EtherChannels. If a StackWise Virtual MEC terminates on a Cisco switch, you can run PAgP protocol on the MEC. If PAgP is running on the MECs between the StackWise Virtual switch and an upstream or downstream switch, the StackWise Virtual can use PAgP to detect a dual-active scenario. The MEC must have at least one port on each switch of the StackWise Virtual setup.

Enhanced PAgP is an extension of the PAgP protocol. In virtual switch mode, ePAgP messages include a new type length value (TLV) which contains the ID of the StackWise Virtual active switch. Only switches in virtual switch mode send the new TLV.

When the StackWise Virtual standby switch detects SVL failure, it initiates SSO and becomes StackWise Virtual active. Subsequent ePAgP messages sent to the connected switch from the newly StackWise Virtual active switch contain the new StackWise Virtual active ID. The connected switch sends ePAgP messages with the new StackWise Virtual active ID to both StackWise Virtual switches.

If the formerly StackWise Virtual active switch is still operational, it detects the dual-active scenario because the StackWise Virtual active ID in the ePAgP messages changes.

Figure 3: Dual-active-detection with ePAgP



Note To avoid PAgP flaps and to ensure that dual-active detection functions as expected, the stack MAC persistent wait timer must be configured as indefinite using the command **stack-mac persistent timer 0 .**

Recovery Actions

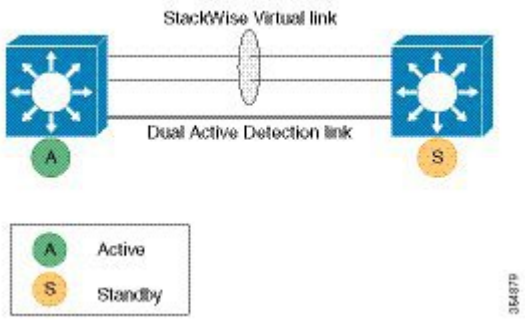
A Cisco StackWise Virtual active switch that detects a dual-active condition shuts down all of its non-SVL or non-DAD interfaces to remove itself from the network. The switch then waits in recovery mode until the SVLs recover. You should physically repair the SVL failure and the switch automatically reloads and restores itself as the standby switch. To enable the switch to remain in recovery mode after restoring the SVL links, see Disabling Recovery Reload, on page 45 section.

Implementing Cisco StackWise Virtual

The two-node solution of Cisco StackWise Virtual is normally deployed at the aggregation layer. Two switches are connected over an SVL.

Cisco StackWise Virtual combines the two switches into a single logical switch with a large number of ports, offering a single point of management. One of the member switches is the active and works as the control and management plane, while the other one is the standby. The virtualization of multiple physical switches into a single logical switch is only from a control and management perspective. Because of the control plane being common, it may look like a single logical entity to peer switches. The data plane of the switches are converged, that is, the forwarding context of a switch might be passed to the other member switch for further processing when traffic is forwarded across the switches. However, the common control plane ensures that all the switches have equivalent data plane entry for each forwarding entity.

Figure 4: Two-Node Solution



An election mechanism that determines which switch is Cisco StackWise Virtual active and which one is a control plane standby, is available. The active switch is responsible for management, bridging and routing protocols, and software data path. These are centralized on the active switch supervisor of the Cisco StackWise Virtual active switch.

How to Configure Cisco StackWise Virtual

Configuring Cisco StackWise Virtual Settings

To enable StackWise Virtual, perform the following procedure :

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**switch** *switch-number* **renumber** *new switch -number* **Example:** `Device#switch 1 renumber 2`	(Optional) Reassigns the switch number. The default switch number will be 1. The valid values for the new switch number are 1 and 2.
Step 3	**switch** *switch-number* **priority** *priority-number* **Example:** `Device#switch 1 priority 5`	(Optional) Assigns the priority number. The default priority number is 1. The highest priority number is 15.

	Command or Action	Purpose
Step 4	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 5	**stackwise-virtual** **Example:** `Device(config)#stackwise-virtual`	Enables Cisco StackWise Virtual and enters stackwise-virtual submode.
Step 6	**domain** *id* **Example:** `Device(config-stackwise-virtual)#domain 2`	(Optional) Specifies the Cisco StackWise Virtual domain ID. The domain ID range is from 1 to 255. The default value is one.
Step 7	**end** **Example:** `Device(config-stackwise-virtual)#end`	Returns to privileged EXEC mode.
Step 8	**show stackwise-virtual** **Example:** `Device#show stackwise-virtual`	
Step 9	**write memory** **Example:** `Device#write memory`	Saves the running-configuration which resides in the system RAM and updates the ROMmon variables. If you do not save the changes, the changes will no longer be part of the startup configuration when the switch reloads. Note that the configurations for **stackwise-virtual** and **domain** are saved to the running-configuration and the startup-configuration after the reload.
Step 10	**reload** **Example:** `Device#reload`	Restarts the switch and forms the stack.

Configuring Cisco StackWise Virtual Link

Note SVL is supported on all 10G, 40G and 25G interfaces of the supported switch models. However, a combination of different interface speeds is not supported.

To configure a switch port as an SVL port, perform the following procedure :

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**interface** { **TenGigabitEthernet** | **FortyGigabitEthernet** | **TwentyFiveGigE** } <*interface*> **Example:** `Device(config)#interface TenGigabitEthernet1/2/0/4`	Enters ethernet interface configuration mode.
Step 4	**stackwise-virtual link** *link value* **Example:** `Device(config-if)#stackwise-virtual link 1`	Associates the interface with configured SVL.
Step 5	**end** **Example:** `Device(config-if)#end`	Returns to privileged EXEC mode.
Step 6	**write memory** **Example:** `Device#write memory`	Saves the running-configuration which resides in the system RAM and updates the ROMmon variables. If you do not save the changes, the changes will no longer be part of the startup configuration when the switch reloads. Note that the configuration for **stackwise-virtual link** *link value* is saved only in the running-configuration and not the startup-configuration.
Step 7	**reload** **Example:** `Device#reload`	Restarts the switch.

Configuring BUM Traffic Optimization

To configure BUM traffic optimization globally, perform the following procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**svl l2bum optimization** **Example:** `Device(config)#svl l2bum optimization`	Enables the BUM traffic optimization within StackWise Virtual setup globally. This feature is enabled by default. Use the **no** form of this command to disable this feature.
Step 4	**end** **Example:** `Device(config-if)#end`	Returns to privileged EXEC mode.
Step 5	**show platform pm l2bum-status vlan** *vlan-id* **Example:** `Device# show platform pm l2bum-status vlan 1`	Displays the number of forwarding ports in VLAN. number of physical ports count in forwarding state
Step 6	**show platform software fed switch ac fss bum-opt summary** **Example:** `Device# show platform software fed switch ac fss bum-opt summary`	Displays the final state of optimization.

Configuring StackWise Virtual Fast Hello Dual-Active-Detection Link

To configure StackWise Virtual Fast Hello DAD link, perform the following procedure. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device>enable`	
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**interface** { **TenGigabitEthernet** | **FortyGigabitEthernet** | **TwentyFiveGigE** } *<interface>* **Example:** `Device(config)#interface TenGigabitEthernet1/2/0/5`	Enters ethernet interface configuration mode.
Step 4	**stackwise-virtual dual-active-detection** **Example:** `Device(config-if)#stackwise-virtual dual-active-detection`	Associates the interface with StackWise Virtual dual-active-detection. **Note** This command will not be visible on the device after the configuration, but will continue to function.
Step 5	**end** **Example:** `Device(config-if)#end`	Returns to privileged EXEC mode.
Step 6	**write memory** **Example:** `Device#write memory`	Saves the running-configuration which resides in the system RAM and updates the ROMmon variables. If you do not save the changes, the changes will no longer be part of the startup configuration when the switch reloads. Note that the configuration for **stackwise-virtual dual-active-detection** is saved only in the running-configuration and not the startup-configuration.
Step 7	**reload** **Example:** `Device#reload`	Restarts the switch and configuration takes effect.

Enabling ePAgP Dual-Active-Detection

To enable ePAgP dual-active-detection on a switch port, perform the following procedure. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**interface** { **TenGigabitEthernet** | **FortyGigabitEthernet** | **TwentyFiveGigE** } *interface* **Example:** `Device(config)#interface TenGigabitEthernet1/2/0/3`	Enters ethernet interface configuration mode.
Step 4	**channel-group** *group_ID* **mode desirable** **Example:** `Device(config-if)#channel-group 1 mode desirable`	Enables PAgP MEC with channel-group id in the range of 1 to 128 for 10 GigabitEthernet interfaces.
Step 5	**exit** **Example:** `Device(config-if)#exit`	Exits interface configuration.
Step 6	**interface port-channel** *channel-group-id* **Example:** `Device(config)#interface port-channel 1`	Selects a port channel interface to configure.
Step 7	**shutdown** **Example:** `Device(config-if)#shutdown`	Shuts down an interface.
Step 8	**exit** **Example:** `Device(config-if)#exit`	Exits interface configuration.
Step 9	**stackwise-virtual** **Example:** `Device(config)#stackwise-virtual`	Enters the StackWise Virtual configuration mode.

	Command or Action	Purpose
Step 10	**dual-active detection pagp** **Example:** `Device(config-stackwise-virtual)#dual-active detection pagp`	Enables pagp dual-active detection. This is enabled by default.
Step 11	**dual-active detection pagp trust channel-group** *channel-group id* **Example:** `Device(config-stackwise-virtual)#dual-active detection pagp trust channel-group 1`	Enables dual-active detection trust mode on channel-group with the configured ID.
Step 12	**exit** **Example:** `Device(config-stackwise-virtual)#exit`	Exits the StackWise-Virtual configuration mode.
Step 13	**interface port-channel** *portchannel* **Example:** `Device(config)#interface port-channel 1`	Configured port-channel on the switch.
Step 14	**no shutdown** **Example:** `Device(config-if)#no shutdown`	Enables the configured port-channel on the switch.
Step 15	**end** **Example:** `Device(config-if)#end`	Exits interface configuration.
Step 16	**write memory** **Example:** `Device#write memory`	Saves the running-configuration which resides in the system RAM and updates the ROMmon variables. If you do not save the changes, the changes will no longer be part of the startup configuration when the switch reloads. Note that the configuration for **dual-active detection pagp trust channel-group** *channel-group id* is saved to the running-configuration and the startup-configuration after the reload.
Step 17	**reload** **Example:** `Device#reload`	Restarts the switch and configuration takes effect.

Disabling Recovery Reload

After recovering from StackWise Virtual link failure, the switch in recovery mode performs a recovery action by automatically reloading the switch . This is the default behaviour in the event of a link failure. In order to retain a switch in recovery mode and prevent the switch from reloading automatically, you must perform the following steps.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**stackwise-virtual** **Example:** `Device(config)#stackwise-virtual`	Enables Cisco StackWise Virtual and enters stackwise-virtual mode.
Step 4	**dual-active recovery-reload-disable** **Example:** `Device(config-stackwise-virtual)#dual-active recovery-reload-disable`	Disables automatic recovery reload of the switch. Note that the configuration for **dual-active recovery-reload-disable** is saved only in the running-configuration and not the startup-configuration.
Step 5	**end** **Example:** `Device(config-stackwise-virtual)#end`	Returns to privileged EXEC mode.

Disabling Cisco StackWise Virtual

To disable Cisco StackWise Virtual on a switch, perform the following procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device>enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**interface** { **TenGigabitEthernet** | **FortyGigabitEthernet** | **TwentyFiveGigE** } *<interface>* **Example:** `Device(config)#interface TenGigabitEthernet 1/2/0/3`	Enters the interface configuration mode.
Step 4	**no stackwise-virtual dual-active-detection** **Example:** `Device(config-if)#no stackwise-virtual dual-active-detection`	Dissociates the interface from StackWise Virtual DAD.
Step 5	Repeat step Step 3, on page 46 **Example:** `Device(config)#interface TenGigabitEthernet 1/2/0/2`	Enters the interface configuration mode.
Step 6	**no stackwise-virtual link** *link* **Example:** `Device(config-if)#no stackwise-virtual link 1`	Dissociates the interface from SVL.
Step 7	**exit** **Example:** `Device(config-if)#exit`	Exits interface configuration.
Step 8	**no stackwise-virtual** **Example:** `Device(config)#no stackwise-virtual`	Disables StackWise Virtual configuration.
Step 9	**exit** **Example:** `Device(config)#exit`	Exits the global configuration mode.
Step 10	**write memory** **Example:**	Saves the running configuration.

	Command or Action	Purpose
	`Device#write memory`	
Step 11	**reload** **Example:** `Device#reload`	Restarts the switch and the configuration takes effect.

Configuration Examples for StackWise Virtual

This section provides the following configuration examples:

- Example: Configuring StackWise Virtual Link, on page 47
- Example: Displaying StackWise Virtual Link Information, on page 48

Example: Configuring StackWise Virtual Link

The following is a sample configuration for configuring SVL on a switch.

On Switch 1:

```
Device>enable
Device#configure terminal
Device(config)#interface TenGigabitEthernet1/0/1
Device(config-if)#stackwise-virtual link 1

WARNING: All the extraneous configurations will be removed for TenGigabitEthernet1/0/1 on
reboot
INFO: Upon reboot, the config will be part of running config but not part of start up config.
Device(config-if)#end
Device#write memory
Device#reload
```

On Switch 2:

```
Device>enable
Device#configure terminal
Device(config)#interface TenGigabitEthernet1/0/1
Device(config-if)#stackwise-virtual link 1

WARNING: All the extraneous configurations will be removed for TenGigabitEthernet1/0/1 on
reboot
INFO: Upon reboot, the config will be part of running config but not part of start up config.
Device(config-if)#end
Device#write memory
Device#reload
```

Example: Configuring StackWise Virtual Fast Hello Dual-Active-Detection Link

The following is a sample configuration for configuring a StackWise Virtual Fast Hello dual-active-detection link on a Switch 1 and Switch 2. You cannot configure StackWise Virtual Fast Hello dual-active-detection links on ports that are already configured as StackWise Virtual link ports.

```
On Switch 1:
Device>enable
Device#configure terminal
Device(config)#interface TenGigabitEthernet3/0/1
Device(config-if)#stackwise-virtual dual-active-detection
WARNING: All the extraneous configurations will be removed for TenGigabitEthernet3/0/1 on
reboot
INFO: Upon reboot, the config will be part of running config but not part of start up config.
Device(config-if)#exit
On Switch 2:
Device(config)#interface TenGigabitEthernet3/0/1
Device(config-if)#stackwise-virtual dual-active-detection
WARNING: All the extraneous configurations will be removed for TenGigabitEthernet3/0/1 on
reboot.
INFO: Upon reboot, the config will be part of running config but not part of start up config.
Device(config-if)#end
On both the switches:
Device#write memory
Device#reload
```

Example: Displaying StackWise Virtual Link Information

Sample output of show stackwise-virtual link command

```
Device#show stackwise-virtual link
Stackwise Virtual Configuration:
--------------------------------------------
Stackwise Virtual : Enabled
Domain Number :            1

Switch  Stackwise Virtual Link   Ports
------  ----------------------   ------
1       1                        TenGigabitEthernet1/1/0/1
2       1                        TenGigabitEthernet2/1/0/1
```

By default in standalone mode, the switches are identified as Switch 1 unless explicitly changed to some other switch number. During the conversion to StackWise Virtual, the switch numbers are changed automatically to reflect two switches in a StackWise Virtual domain.

Example:DisplayingStackWiseVirtualDual-Active-DetectionLinkInformation

Sample output of show stackwise-virtual dual-active-detection command

StackWise Virtual DAD links configuration:

```
Device#show stackwise-virtual dual-active-detection
Recovery Reload for switch 1: Enabled
Recovery Reload for switch 2: Enabled
```

```
Dual-Active-Detection Configuration:
-------------------------------------
Switch   Dad port                          Status
------   ------------                      ---------
1        TenGigabitEthernet1/3/0/1         up
2        TenGigabitEthernet2/3/0/1         up
```

StackWise Virtual DAD links configuration after configuring the **dual-active recovery-reload-disable** command:

```
Device#show stackwise-virtual dual-active-detection
Recovery Reload for switch 1: Enabled
Recovery Reload for switch 2: Enabled
Dual-Active-Detection Configuration:
-------------------------------------
Switch      Dad port                       Status
------      ------------                   ---------
1           TenGigabitEthernet1/3/0/1      up
2           TenGigabitEthernet2/3/0/1      up
```

Sample output of show stackwise-virtual dual-active-detection epagp command

StackWise Virtual DAD ePAgP information:

```
Device#show stackwise-virtual dual-active-detection pagp
Pagp dual-active detection enabled: Yes
In dual-active recovery mode: No
Recovery Reload for switch 1: Enabled
Recovery Reload for switch 2: Enabled

Channel group 11
                  Dual-Active           Partner              Partner         Partner
Port              Detect Capable        Name                 Port            Version
Fo1/1/0/17        Yes                   SwitchA              Hu2/0/1         1.1
Fo2/2/0/21        Yes                   SwitchA              Hu1/0/4         1.1
```

Partner Name and **Partner Port** fields in the output represent the name and the ports of the peer switch to which the PagP port-channel is connected through MEC.

Verifying Cisco StackWise Virtual Configuration

To verify your StackWise Virtual configuration, use the following **show** commands:

show stackwise-virtual switch *number <1-2>*	Displays information of a particular switch in the stack.
show stackwise-virtual link	Displays StackWise Virtual link information.
show stackwise-virtual bandwidth	Displays the bandwidth available for the Cisco StackWise Virtual.
show stackwise-virtual neighbors	Displays the Cisco StackWise Virtual neighbors.

show stackwise-virtual dual-active-detection	Displays StackWise Virtual dual-active-detection information.
show stackwise-virtual dual-active-detection pagp	Displays ePAgP dual-active-detection information.
Switch ½ renumber ½	(Optional)Assigns a new switch number. The default number is 1.

Additional References for StackWise Virtual

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	High Availability Command Reference for Catalyst 9400 Switches

Feature History for Cisco StackWise Virtual

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 Fuji 16.9.1	Cisco StackWise Virtual	Cisco StackWise Virtual is a network system virtualization technology that pairs two switches into one virtual switch to simplify operational efficiency with a single control and management plane.
		Support for this feature was introduced on the following chassis models and supervisor modules:
		• Chassis Models: Catalyst 9404R Switch and Catalyst 9407R Switch.
		• Supervisor Modules:
		  • Cisco Catalyst 9400 Series Supervisor 1 Module (C9400-SUP-1). The feature requires a special, additional, C9400-SUP-UPG-LIC= license with this supervisor module.
		  • Cisco Catalyst 9400 Series Supervisor 1XL Module(C9400-SUP-1XL).

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Recovery Reload	Support for disabling DAD recovery reload was introduced. Enter the **dual-active recovery-reload-disable** command in stackwise virtual mode (config-stackwise-virtual).
	Cisco StackWise Virtual: Catalyst 9410R Switch	Support for this feature was introduced on the Catalyst 9410R Switch.
Cisco IOS XE Amsterdam 17.2.1	BUM Traffic Optimization on switches with Cisco StackWise Virtual	Support for the *BUM Traffic Optimization* feature was introduced on switches with Cisco StackWise Virtual configured.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/.