CHAPTER 1

Configuring Virtual Switching Systems

This chapter describes how to configure a virtual switching system (VSS) for the Catalyst 4500/4500X series switch (Supervisor Engine 7-E, Supervisor Engine 7L-E, and Catalyst 4500-X). Cisco Release IOS XE 3.4.0SG and later releases support VSS.

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

- The *Cisco IOS Virtual Switch Command Reference* at this URL:

  http://www.cisco.com/en/US/docs/ios/vswitch/command/reference/vs_book.html
- The Cisco IOS Software Release 12.4 Mainline at this URL:

  http://www.cisco.com/en/US/products/ps6350/prod_command_reference_list.html

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

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

If a command is not in the *Catalyst 4500 Series Switch Command Reference*, you can locate it in the Cisco IOS library. See the *Cisco IOS Master Command List, Release 12.2SX* and related publications at this location:

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

This chapter consists of these sections:

Understanding Virtual Switching Systems

These sections describe a VSS:

- VSS Overview, page 1-2
- VSS Redundancy, page 1-11
- Multichassis EtherChannels, page 1-14
- Packet Handling, page 1-16
- System Monitoring, page 1-20
- Dual-Active Detection, page 1-23
- Configuring a Recovery IP Address, page 1-25
- VSS Initialization, page 1-26

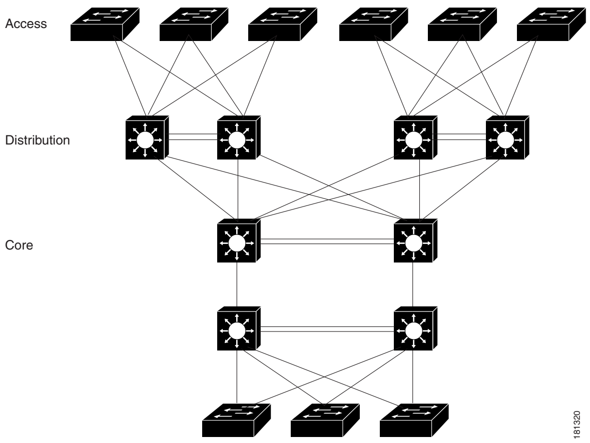
VSS Overview

Network operators increase network reliability by configuring switches and by provisioning links to the redundant pairs. Figure 1-1 shows a typical switch network configuration. Redundant network elements and redundant links can add complexity to network design and operation. Virtual switching simplifies the network by reducing the number of network elements and hiding the complexity of managing redundant switches and links.

A VSS combines a pair of Catalyst 4500 or 4500-X series switches into a single network element. The VSS manages the redundant links, which externally act as a single port channel.

The VSS simplifies network configuration and operation by reducing the number of Layer 3 routing neighbors and by providing a loop-free Layer 2 topology.

Figure 1-1 Typical Switch Network Design



The following sections present an overview of the VSS. These topics are covered in detail in subsequent chapters:

- Key Concepts, page 1-3
- VSS Functionality, page 1-5

- Hardware Requirements, page 1-9
- Understanding VSL Topology, page 1-11

Key Concepts

The VSS incorporates the following key concepts:

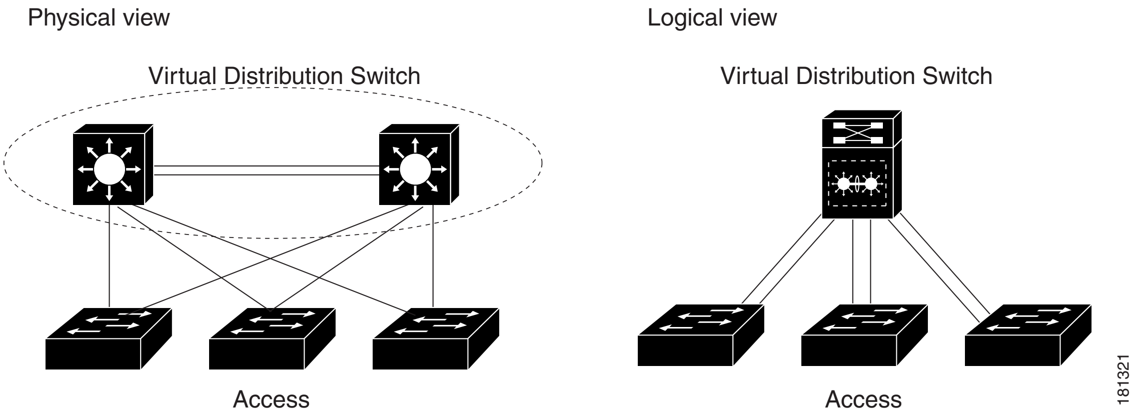
- Virtual Switching System, page 1-3
- VSS Active and VSS Standby Switch, page 1-3
- Virtual Switch Link, page 1-4
- Multichassis EtherChannel, page 1-5

Virtual Switching System

A VSS combines a pair of switches into a single network element. For example, a VSS in the distribution layer of the network interacts with the access and core networks as if it were a single switch. See Figure 1-2.

An access switch connects to both switches of the VSS using one logical port channel. The VSS manages redundancy and load balancing on the port channel. This capability enables a loop-free Layer 2 network topology. The VSS also simplifies the Layer 3 network topology by reducing the number of routing peers in the network.

Figure 1-2 VSS in the Distribution Network



VSS Active and VSS Standby Switch

When you create or restart a VSS, the peer switches negotiate their roles. One switch becomes the VSS Active switch, and the other switch becomes the VSS Standby switch.

The VSS Active controls the VSS, running the Layer 2 and Layer 3 control protocols for the switching modules on both switches. The VSS Active switch also provides management functions for the VSS, such as module online insertion and removal (OIR) and the console interface.

The VSS Active and VSS Standby switches perform packet forwarding for ingress data traffic on their locally hosted interfaces. However, the VSS Standby switch sends all control traffic to the VSS Active switch for processing.

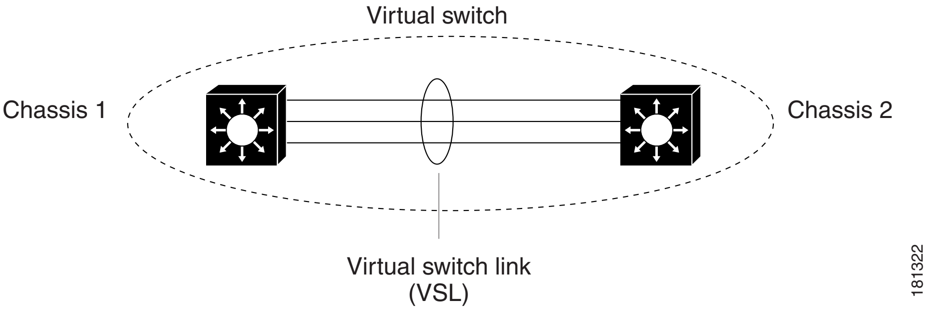
Virtual Switch Link

For the two switches of the VSS to act as one network element, they need to share control information and data traffic.

The virtual switch link (VSL) is a special link that carries control and data traffic between the two switches of a VSS, as shown in Figure 1-3. The VSL is implemented as an EtherChannel with up to eight links. The VSL gives control and management traffic higher priority than data traffic so that control and management messages are never discarded. Data traffic is load balanced among the VSL links by the EtherChannel load-balancing algorithm.

Note EtherChannel load balancing method is a global configuration; VSL observes that method of load balancing.

Figure 1-3 Virtual Switch Link



When you configure VSL, all existing configurations are removed from the interface except for specific allowed commands. When you configure VSL, the system puts the interface into a restricted mode. This means that only specific configuration commands can be configured on the interface.

The following VSL configuration commands are inserted automatically on all VSL member ports:

- **switchport mode trunk**
- **switchport nonegotiate**
- **no lldp transmit**
- **no lldp receive**
- **no cdp enable**
- **service-policy output VSL-Queuing-Policy**

In VSL restricted mode, only these configuration commands are available:

- **channel-group**
- **default**
- **description**
- **exit**
- **load-interval**
- **logging**
- **no**
- **power**

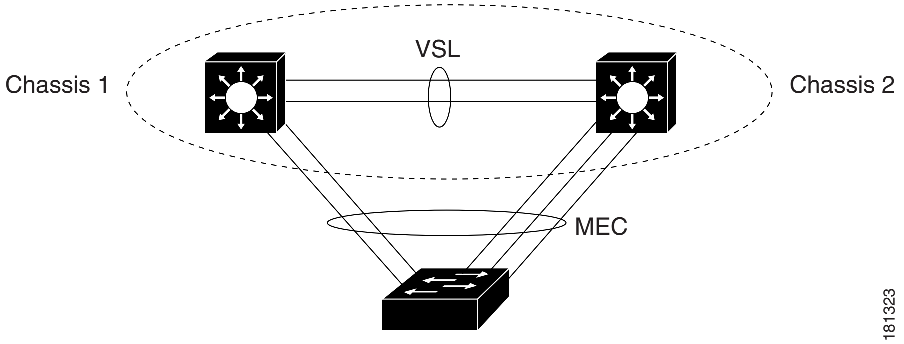
- **service-policy**
- **shutdown**

Multichassis EtherChannel

Note Beginning with Cisco Release IOS XE 3.5.0E and IOS 15.2(1)E, Layer 3 MEC is supported on the Catalyst 4500 series switch. Cisco Release IOS XE 3.4.0SG does not support Layer 3 MEC.

An EtherChannel (also known as a port channel) is a collection of two or more physical links that combine to form one logical link. Layer 2 protocols operate on the EtherChannel as a single logical entity. A VSS enables the creation of Multi-Chassis EtherChannel (MEC), which is an Etherchannel whose member ports can be distributed across the member switches in a VSS. Because non-VSS switches connected to a VSS view the MEC as a standard EtherChannel, non-VSS switches can connect in a dual homed manner. Figure 1-4 displays a dual-homed connection for an MEC into the VSS; VSS is seen as a single logical switch. Traffic traversing an MEC can be load balanced locally within a VSS member switch much as in standard EtherChannels. Cisco MEC supports the bundling protocols LACP and PAgP as well as ON mode.

Figure 1-4 VSS with MEC



VSS supports a maximum of 256 EtherChannels. This limit applies to the total number of regular EtherChannels and MECs. Because the VSL requires two EtherChannel numbers (one for each switch in the VSS), there are 254 user-configurable EtherChannels.

For information on how to configure Layer 3 Multichassis EtherChannels, see For information on how to configure Layer 3 Multichassis EtherChannels, see, page 1-5

VSS Functionality

The following sections describe the main functionality of a VSS:

- Redundancy and High Availability, page 1-6
- Packet Handling, page 1-6

- System Management, page 1-6
- Quad-Supervisor (In-chassis Standby Supervisor Engine) Support, page 1-6
- Asymmetric chassis support, page 1-8
- Interface Naming Convention, page 1-8
- Module Number Convention, page 1-8
- Key Software Features not Supported on VSS, page 1-8

Redundancy and High Availability

In a VSS, supervisor engine redundancy operates between the VSS Active and VSS Standby switch, using stateful switchover (SSO) and nonstop forwarding (NSF). The peer switch exchange configuration and state information across the VSL and the VSS Standby supervisor engine runs in SSO-HOT mode.

The VSS Standby switch monitors the VSS Active switch using the VSL. If it detects failure, the VSS Standby switch initiates a switchover and takes on the VSS Active role. When the failed switch recovers, it takes on the VSS Standby role.

If either the VSS Active switch fails or all links that belong to the VSL port-channel fail, the VSS Standby switch initiates a switchover and assumes the role of the VSS Active switch. If the previous VSS Active switch has failed, it reloads and boots as the VSS Standby switch. However, if only the VSL port-channel failure caused the switchover, the previous VSS Active switch enters recovery mode (provided dual-active detection is configured). In this scenario, the previous VSS Active chassis (now in recovery mode) carries no traffic and only monitors the VSL link. When one link in the VSL port-channel is up, the recovery mode switch reloads and boots as a VSS Standby chassis. For additional information about dual-active detection, see the "Dual-Active Detection" section on page 1-23.

Packet Handling

The VSS Active supervisor engine runs the Layer 2 and Layer 3 protocols and features for the VSS and manages all ports on both switches.

The VSS uses VSL to communicate protocol and system information between the peer switches and to carry data traffic between the switches when required.

Both switches perform packet forwarding for ingress traffic on their interfaces. If possible, ingress traffic is forwarded to an outgoing interface on the same switch to minimize data traffic that must traverse the VSL.

System Management

The VSS Active supervisor engine acts as a single point of control for the VSS. For example, the VSS Active supervisor engine handles OIR of switching modules on both switches. The VSS Active supervisor engine uses VSL to send messages to and from local ports on the VSS Standby switch.

The command console on the VSS Active supervisor engine is used to control both switches. In virtual switch mode, the command console on the VSS Standby supervisor engine blocks attempts to enter configuration mode.

The VSS Standby switch runs a subset of system management tasks. For example, the VSS Standby switch handles its own power management, linecard bringup, and other local hardware management.

Quad-Supervisor (In-chassis Standby Supervisor Engine) Support

The Catalyst 4500 series switches support dual supervisors in a redundant chassis, which can be configured for SSO or RPR mode. However, when a chassis is running in VSS mode, it supports a second supervisor engine, but only in rommon mode. In-Chassis-Standby (ICS) can not participate in control, management, or forwarding plane functioning. This makes ports on the supervisor engine in rommon mode available for forwarding although it neither participates in any switchover nor provides protection against any failure. In VSS mode, an In-Chassis-Active (ICA) supervisor engine participates in VSS control/ management operation and manages ports on the supervisor engine in rommon mode.

If the second supervisor engine is inserted in a redundant chassis, the following information applies:

- It must also be manually configured for VSS mode, i.e., it must have been converted from standalone to VSS mode previously. If you insert a supervisor engine that was not configured for VSS mode, it will disrupt the operation of the ICA supervisor engine. If it was previously configured, automatic boot must be disabled (i.e., to boot only to ROM Monitor) with the **confreg** command in rommon.

  The supervisor engine does not takeover or boot automatically when the ICA supervisor engine fails. A manual boot up is required to make it participate in VSS; it then functions as an ICA supervisor engine.

  More details on rommon commands are found at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/15.1.2/XE_340/configuration/guide/rommon.html#wp1013959

- A supervisor engine's conversion from standalone to VSS occurs per engine. If two supervisor engines exist in a chassis, one should be retained in rommon or removed, before conversion occurs. You can convert the second supervisor to VSS mode while the first supervisor is removed or in rommon, with the additional step of setting it to “boot only to ROM Monitor.” When both engines are converted to VSS, they can be inserted into the chassis together and re-booted.
- Booting a chassis with two supervisor engines configured for VSS causes one of the engines to become the ICA and participate in VSS. The other engine, which becomes the ICS, will continuously reload. The secondary supervisor (the ICS) must be configured to “boot only to ROM Monitor” with automatic boot disabled.
- When the ICA fails, the ICS doesn't take over because ICS support of SSO or RPR mode is unavailable. ICS (the secondary supervisor) must be booted manually to become the ICA and manage the VSS operations. For this to happen, the former active supervisor engine must remain in rommon mode.
- ISSU support requires ICA supervisor engines on both chassis. The ICS supervisor engine does not participate in upgrade or any forwarding operations.
- Because ICS supervisor engines do not communicate with ICA supervisors, VSS and other configurations must be done at conversion time on the ICS. If not done or the configurations do not match the necessary VSS parameters (like, SwitchId, Domain, and VSL configurations), it cannot form a VSS when ICA goes down and ICS is booted manually. You can, however, enter these “bootup” commands to make it join an existing VSS domain.

Note When a supervisor engine in VSS mode is booting in a chassis, where an ICA supervisor engine already exists, the ICS supervisor engine (the one that is booting) is continuously reset. It must be manually put into rommon by disabling auto-boot. Simultaneously, the ICS may display a message that it has crashed and might generate a crashdump. Because the supervisor engine is going down, this message is harmless; it does not affect the functionality of VSS. Instead of resetting itself gracefully, the engine might crash while attempting a reset.

Asymmetric chassis support

Catalyst 4500 and Catalyst 4500X VSS require the same supervisor engine type in both chassis. The chassis can differ in type (i.e., +E and -E chassis can be in a single VSS) and also can differ in the number of slots in chassis. VSS cannot be formed between different flavors of Catalyst 4500X (e.g., 4500X-16 and 4500X-32).

Interface Naming Convention

In VSS mode, interfaces are specified using the switch number (in addition to slot and port), because the same slot numbers are used on both chassis. For example, the **interface 1/5/4** command specifies port 4 of the switching module in slot 5 of switch 1. The **interface 2/5/4** command specifies port 4 on the switching module in slot 5 of switch 2.

Module Number Convention

IOS treats modules in both chassis as if they belong to one single chassis and the module number space is 1-20.

Switch 1 receives a module number from 1-10 and switch 2 receives a number from 11-20, irrespective the chassis type, supervisor type, or number of slots in a chassis. For example, on a 3-slot chassis VSS, the module numbers on switch 1 would be 1, 2, and 3, and on switch 2, the numbers would be 11, 12, and 13. The module number on switch 2 always starts from 11.

The **show switch virtual slot-map** command provides virtual to physical slot mapping. The following is a sample output:

```
Virtual    Remote      Physical   Module
Slot No    Switch No   Slot No    Uptime
---------+-----------+----------+----------
  1         1              1        00:24:14
  2         1              2        00:23:46
  3         1              3          -
  4         1              4          -
  5         1              5          -
  6         1              6          -
  7         1              7          -
  8         1              8          -
  9         1              9          -
  10        1              10         -
  11        2              1        00:22:03
  12        2              2        00:24:43
  13        2              3        00:24:43
  14        2              4          -
  15        2              5          -
  16        2              6          -
  17        2              7          -
  18        2              8          -
  19        2              9          -
  20        2              10         -
```

Key Software Features not Supported on VSS

With some exceptions, the VSS maintains feature parity with the standalone Catalyst 4500 or 4500-X series switches. Major exceptions include:

- CFM D8.1
- Dot1q Tunnel (**“legacy/classic”** dot1q tunnel)
- Dot1q tunneling and L2PT (Layer 2 Protocol Tunneling)

- Energywise
- Fast UDLD
- Flexlink
- Mediatrace (Medianet active video monitoring feature)
- Metadata (Medianet feature)
- Per VLAN Learning
- REP and associated featurettes
- UDE
- UDLR
- VLAN Translation (1:1 and 1:2-Selective QinQ)
- VMPS Client
- WCCP

Hardware Requirements

The following sections describe the hardware requirements of a VSS:

- Chassis and Modules, page 1-9
- VSL Hardware Requirements, page 1-10
- Multichassis EtherChannel Requirements, page 1-11

Chassis and Modules

Table 1-1 describes the hardware requirements for the VSS chassis and modules.

Table 1-1 VSS Hardware Requirements

Hardware	Count	Requirements
Chassis	2	VSS is available on a Catalyst 4500-X switch and on chassis that support Supervisor Engine 7-E and Supervisor Engine 7-LE. **Note** +E and -E chassis can be mixed.

Table 1-1 VSS Hardware Requirements

Hardware	Count	Requirements
Supervisor Engines	2	VSS is available on Supervisor Engine 7-E, Supervisor Engine 7L-E, and on the Catalyst 4500-X switch series. All supervisor engines or systems in a VSS must match precisely.
Linecard	0 to as many linecard slots are available in a chassis.	WS-X4748-SFP-E WS-X4724-SFP-E WS-X4712-SFP-E WS-X4748-RJ45V+E WS-X4712-SFP+E WS-X4640-CSFP-E WS-X4748-UPOE+E WS-X4748-RJ45-E WS-X4606-X2-E WS-X4648-RJ45V-E WS-X4648-RJ45V+E WS-X4648-RJ45-E WS-X4624-SFP-E WS-X4612-SFP-E WS-X4548-RJ45V+ WS-X4448-GB-SFP WS-X4306-GB WS-X4248-RJ45V WS-X4248-FE-SFP WS-X4148-RJ WS-X4148-FX-MT

VSL Hardware Requirements

The VSL EtherChannel supports both 10-Gigabit Ethernet ports and 1- Gigabit Ethernet ports.

We recommend that you use at least two of the 10-Gigabit/1-Gigabit Ethernet ports to create the VSL between the two switches. You cannot combine 10-Gigabit and 1-Gigabit Ethernet ports in a VSL port-channel.

Be aware of the following:

- You can add additional physical links to the VSL EtherChannel with the 10-Gigabit Ethernet ports on any supported supervisor engine or linecard.
- Oversubscribed linecard ports can be used for VSL but total bandwidth requirements of VSL or any traffic drop because of a certain hashing mechanism must be accounted for before using oversubscribed linecard ports for VSL.

- VSL ports can have only 10 Gigabit Ethernet port mode on a WS-X4606-X2-E linecard; non-VSL ports can be configured as 10 or 1 Gigabit Ethernet port mode.
- 1 Gigabit Ethernet ports on line card X4606-X2-E cannot be used as VSL links.

Multichassis EtherChannel Requirements

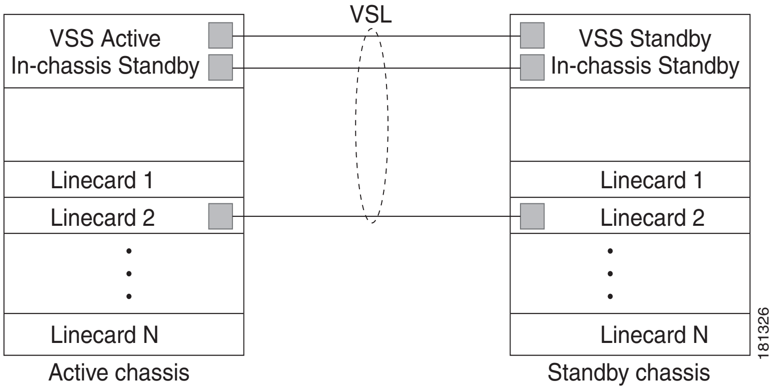
Physical links from any of the supervisor engines or linecard modules can be used to implement a Multichassis EtherChannel (MEC).

Understanding VSL Topology

A VSS contains two switches that communicate using the VSL, which is a special port group.

We recommend that you configure at least two of the 10-Gigabit/1-Gigabit Ethernet ports as VSL, selecting ports from different modules. Figure 1-5 shows a example topology.

Figure 1-5 VSL Topology Example



VSS Redundancy

The following sections describe how redundancy in a VSS supports network high availability:

- Overview, page 1-12
- RPR and SSO Redundancy, page 1-12
- Switch Roles in a VSS, page 1-12
- Failed Switch Recovery, page 1-13
- VSL Failure, page 1-14
- User Actions, page 1-14

Overview

A VSS operates stateful switchover (SSO) between the VSS Active and VSS Standby supervisor engines. Compared to standalone mode, a VSS has the following important differences in its redundancy model:

- The VSS Active and VSS Standby supervisor engines are hosted in separate switches and use the VSL to exchange information.
- The VSS Active supervisor engine controls both switches of the VSS. The VSS Active supervisor engine runs the Layer 2 and Layer 3 control protocols and manages the switching modules on both switches.
- The VSS Active and VSS Standby switches perform data traffic forwarding.

If the VSS Active supervisor engine fails, the VSS Standby supervisor engine initiates a switchover and assumes the VSS Active role.

RPR and SSO Redundancy

A VSS operates with stateful switchover (SSO) redundancy if it meets the following requirements:

- Both supervisor engines must be running the same software version, unless it is in the process of software upgrade.
- VSL-related configuration in the two switches must match.
- SSO and nonstop forwarding (NSF) must be configured on each switch.

Note See the "SSO Dependencies" section on page 1-27 for additional details about the requirements for SSO redundancy on a VSS. See Chapter 1, "Configuring Cisco NSF with SSO Supervisor Engine Redundancy" for information about configuring SSO and NSF.

With SSO redundancy, the VSS Standby supervisor engine is always ready to assume control following a fault on the VSS Active supervisor engine. Configuration, forwarding, and state information are synchronized from the VSS Active supervisor engine to the redundant supervisor engine at startup and whenever changes to the VSS Active supervisor engine configuration occur. If a switchover occurs, traffic disruption is minimized.

If a VSS does not meet the requirements for SSO redundancy, it will be incapable of establishing a relationship with the peer switch. Catalyst 4500/4500-X series switches' VSS does not support route processor redundancy (RPR) mode.

The VSS runs stateful switchover (SSO) between the VSS Active and VSS Standby supervisor engines (see Figure 1-6). The VSS determines the role of each supervisor engine during initialization.

The supervisor engine in the VSS Standby switch runs in hot standby state. The VSS uses the VSL link to synchronize configuration data from the VSS Active to the VSS Standby supervisor engine. Also, protocols and features that support high availability synchronize their events and state information to the VSS Standby supervisor engine.

Switch Roles in a VSS

Figure 1-6 illustrates the switches' roles in a VSS.

Figure 1-6 Switches' Roles in a VSS

Failed Switch Recovery

If the VSS Active switch or supervisor engine fails, the VSS initiates a stateful switchover (SSO) and the former VSS Standby supervisor engine assumes the VSS Active role. The failed switch performs recovery action by reloading the supervisor engine.

If the VSS Standby switch or supervisor engine fails, no switchover is required. The failed switch performs recovery action by reloading the supervisor engine.

The VSL links are unavailable while the failed switch recovers. After the switch reloads, it becomes the new VSS Standby switch and the VSS reinitializes the VSL links between the two switches.

The switching modules on the failed switch are unavailable during recovery, so the VSS operates only with the MEC links that terminate on the VSS Active switch. The bandwidth of the VSS is reduced until the failed switch has completed its recovery and become operational again. Any devices that are connected only to the failed switch experience an outage.

Note The VSS may experience a brief data path disruption when the switching modules in the VSS Standby switch become operational after the SSO.

After the SSO, much of the processing power of the VSS Active supervisor engine is consumed in bringing up a large number of ports simultaneously in the VSS Standby switch. As a result, some links might be brought up before the supervisor engine has configured forwarding for the links, causing traffic to those links to be lost until the configuration is complete. This condition is especially disruptive if the link is an MEC link and it is running in "ON" mode. This is why it is recommended that MEC ports always have either PAgP or LACP mode of EtherChannel configured.

Note We recommend not configuring LACP independent mode (standalone-mode) for MEC because ports on the VSS Standby switch (while it boots) come up tens of seconds before the control plane is fully functional. This behavior causes a port to start working in independent mode and might cause traffic loss until the port is bundled.

VSL Failure

To ensure fast recovery from VSL failures, fast link failure detection is enabled in virtual switch mode on all VSL port channel members.

Note Fast link notification is based upon internal hardware assisted BFD sessions between the pair of physical VSL links.

If a single VSL physical link goes down, the VSS adjusts the port group so that the failed link is not selected.

If the VSS Standby switch detects complete VSL link failure, it initiates a stateful switchover (SSO). If the VSS Active switch has failed (causing the VSL links to go down), the scenario is switch failure, as described in the previous section.

If only the VSL has failed and the VSS Active switch is still operational, this is a dual-active scenario. The VSS detects that both switches are operating in VSS Active mode and performs recovery action. See the "Dual-Active Detection" section on page 1-23 for additional details about the dual-active scenario.

User Actions

From the VSS Active switch command console, you can initiate a VSS switchover or a reload.

If you enter the **reload** command from the command console, it performs a reload on the switch where reload is issued.

To reload only the VSS Standby switch, use the **redundancy reload peer** command.

To force a switchover from the VSS Active to the VSS Standby supervisor engine, use the **redundancy force-switchover** command.

To reset both the VSS Active and Standby switch, use the **redundancy reload shelf** command.

Multichassis EtherChannels

These sections describe multichassis EtherChannels (MECs):

- Overview, page 1-14
- MEC Failure Scenarios, page 1-15

Overview

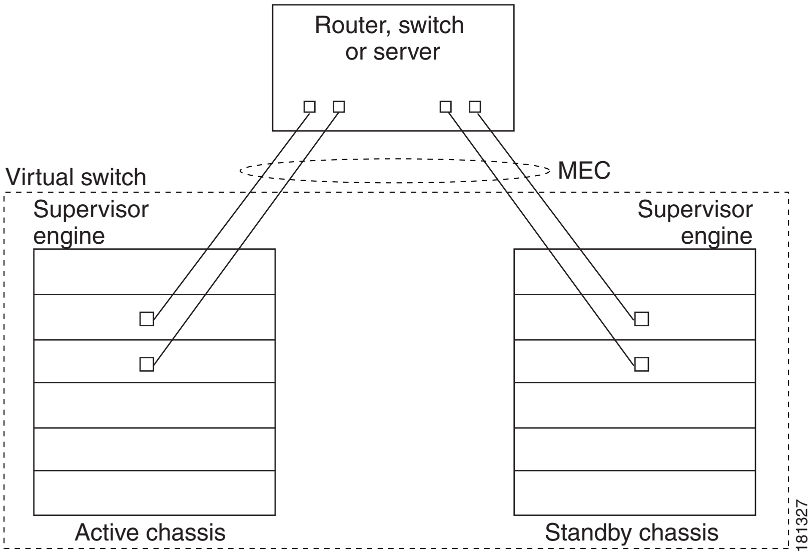
A multichassis EtherChannel is an EtherChannel with ports that terminate on both switches of the VSS (see Figure 1-7). A VSS MEC can connect to any network element that supports EtherChannel (such as a host, server, router, or switch).

At the VSS, an MEC is an EtherChannel with additional capability: the VSS balances the load across ports in each switch independently. For example, if traffic enters the VSS Active switch, the VSS will select an MEC link from the VSS Active switch. This MEC capability ensures that data traffic does not unnecessarily traverse the VSL.

Each MEC can optionally be configured to support either PAgP or LACP. These protocols run only on the VSS Active switch. PAgP or LACP control packets destined for an MEC link on the VSS Standby switch are sent across VSL.

An MEC can support up to eight physical links, which can be distributed in any proportion between the VSS Active and VSS Standby switch.

Figure 1-7 MEC Topology



MEC Failure Scenarios

We recommend that you configure the MEC with at least one link to each switch. This configuration conserves VSL bandwidth (traffic egress link is on the same switch as the ingress link), and increases network reliability (if one VSS supervisor engine fails, the MEC is still operational).

The following sections describe possible failures and the resulting impacts:

- Single MEC Link Failure, page 1-15
- All MEC Links to the VSS Active Switch Fail, page 1-15
- All MEC Links to the VSS Standby Switch Fail, page 1-16
- All MEC Links Fail, page 1-16
- VSS Standby Switch Failure, page 1-16
- VSS Active Switch Failure, page 1-16

Single MEC Link Failure

If a link within the 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 VSS Active Switch Fail

If all links to the VSS Active switch fail, the MEC becomes a regular EtherChannel with operational links to the VSS Standby switch.

Data traffic terminating on the VSS Active switch reaches the MEC by crossing the VSL to the VSS Standby switch. Control protocols continue to run in the VSS Active switch. Protocol messages reach the MEC by crossing the VSL.

All MEC Links to the VSS Standby Switch Fail

If all links fail to the VSS Standby switch, the MEC becomes a regular EtherChannel with operational links to the VSS Active switch.

Control protocols continue to run in the VSS Active switch. All control and data traffic from the VSS Standby switch reaches the MEC by crossing the VSL to the VSS Active switch.

All MEC Links Fail

If all 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 Spanning Tree Protocol (STP) perform the same corrective action as for a regular EtherChannel.

VSS Standby Switch Failure

If the VSS Standby switch fails, the MEC becomes a regular EtherChannel with operational links on the VSS Active switch. Connected peer switches detect the link failures, and adjust their load-balancing algorithms to use only the links to the VSS Active switch.

VSS Active Switch Failure

VSS Active switch failure results in a stateful switchover (SSO). See the "VSS Redundancy" section on page 1-11 for details about SSO on a VSS. After the switchover, the MEC is operational on the new VSS 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 VSS Active switch.

Packet Handling

In a VSS, the VSS Active supervisor engine runs the Layer 2 and Layer 3 protocols and features for the VSS and manages the ports on both switches.

The VSS uses the VSL to communicate system and protocol information between the peer switches and to carry data traffic between the two switches.

Both switches perform packet forwarding for ingress traffic on their local interfaces. The VSS minimizes the amount of data traffic that must traverse the VSL.

The following sections describe packet handling in a VSS:

- Traffic on the VSL, page 1-16
- Layer 2 Protocols, page 1-17
- Layer 3 Protocols, page 1-18

Traffic on the VSL

The VSL carries data traffic and in-band control traffic between the two switches. All frames forwarded over the VSL link are encapsulated with a special header (up to ten bytes for data traffic and 18 bytes for control packets), which provides information for the VSS to forward the packet on the peer switch.

The VSL transports control messages between the two switches. Messages include protocol messages that are processed by the VSS Active supervisor engine, but received or transmitted by interfaces on the VSS Standby switch. Control traffic also includes module programming between the VSS Active supervisor engine and switching modules on the VSS Standby switch.

The VSS needs to transmit data traffic over the VSL under the following circumstances:

- Layer 2 traffic flooded over a VLAN (even for dual-homed links).
- Packets processed by software on the VSS Active supervisor engine where the ingress interface is on the VSS Standby switch.
- The packet destination is on the peer switch, such as 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.

VSL also transports system data, such as NetFlow export data and SNMP data, from the VSS Standby switch to the VSS Active supervisor engine.

To preserve the VSL bandwidth for critical functions, the VSS uses strategies to minimize user data traffic that must traverse the VSL. For example, if an access switch is dual-homed (attached with an MEC terminating on both VSS switches), the VSS transmits packets to the access switch using a link on the same switch as the ingress link.

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

Layer 2 Protocols

The VSS Active supervisor engine runs the Layer 2 protocols (such as STP and VTP) for the switching modules on both switches. Protocol messages that are transmitted and received on the VSS Standby switch switching modules must traverse the VSL to reach the VSS Active supervisor engine.

All Layer 2 protocols in VSS work similarly in standalone mode. The following sections describe the difference in behavior for some protocols in VSS:

- Spanning Tree Protocol, page 1-17
- EtherChannel Control Protocols, page 1-18
- Jumbo frame size restriction, page 1-18
- SPAN, page 1-18
- Private VLANs, page 1-18

Spanning Tree Protocol

The VSS Active switch runs Spanning Tree Protocol (STP). The VSS Standby switch redirects STP BPDUs across the VSL to the VSS Active switch.

The STP bridge ID is commonly derived from the chassis MAC address. To ensure that the bridge ID does not change after a switchover, the VSS continues to use the original chassis 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. The VSS defines a common device identifier for both chassis. You should use PAgP or LACP on MECs instead of mode ON, although all three modes are supported.

A new PAgP enhancement has been defined for assisting with dual-active scenario detection. For additional information, see the "Dual-Active Detection" section on page 1-23.

Jumbo frame size restriction

The maximum jumbo frame size supported on a VSS interface is 9188 bytes (MTU of 9170 bytes). This accommodates the overhead of transporting packets between the two member switches over VSL.

Not all frames traverse VSL. So, packets confined to one of the member switches could have a size of 9216 bytes (MTU of 9198 bytes). Such frames may require diversion over VSL when a failure occurs. This is why the *max configured MTU* on non-VSL front panel ports is 9170.

Note The MTU CLI is unavailable on a VSL interface. It is set internally to 9198 (Max frame size of 9216), addressing the overhead of VSL.

For example, if we send traffic between two ports on the active switch, no overhead exists. However, overhead exists when we send packets between ports of active to ports of standby. Even more overhead exists when we send packets from standby ports to the active CPU. The higher limit accommodates the worst case and guarantees consistent forwarding under all scenarios.

SPAN

VSS supports all SPAN features for non-VSL interfaces.

Note SPAN on VSL ports is not supported; VSL ports can be neither a SPAN source, nor a SPAN destination.

The number of SPAN sessions available on a VSS matches that on a single switch running in standalone mode.

Private VLANs

Private VLANs on VSS work similarly in standalone mode. The only exception is that the native VLAN on isolated trunk ports must be configured explicitly. Refer to Chapter 1, "Configuring Private VLANs" for details on how to configure the native VLAN on isolated trunk ports.

Layer 3 Protocols

The VSS Active supervisor engine runs the Layer 3 protocols and features for the VSS. All layer 3 protocol packets are sent to and processed by the VSS Active supervisor engine. Both member switches perform hardware forwarding for ingress traffic on their interfaces. If possible, to minimize data traffic that must traverse the VSL, ingress traffic is forwarded to an outgoing interface on the same switch. When software forwarding is required, packets are sent to the VSS Active supervisor engine for processing.

The same router MAC address, assigned by the VSS Active supervisor engine, is used for all Layer 3 interfaces on both VSS member switches. After a switchover, the original router MAC address is still used. The router MAC address is configurable and can be chosen from three options: virtual-mac (derived from domainId), chassis-mac (preserved after switchover), and user-configured MAC address. VSS uses virtual MAC address as the default.

The following sections describe Layer 3 protocols for a VSS:

- IPv4, page 1-19
- IPv6, page 1-19
- IPv4 Multicast, page 1-19
- Software Features, page 1-20

IPv4

The supervisor engine on the VSS Active switch runs the IPv4 routing protocols and performs any required software forwarding. All routing protocol packets received on the VSS Standby switch are redirected to the VSS Active supervisor engine across the VSL. The VSS Active supervisor engine generates all routing protocol packets to be sent out over ports on either VSS member switch.

Hardware forwarding is distributed across both members on the VSS. The supervisor engine on the VSS Active switch sends Forwarding Information Base (FIB) updates to the VSS Standby supervisor engine, which installs all routes and adjacencies in its 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 VSL.

The supervisor engine on the VSS Active switch performs all software forwarding (for protocols such as IPX) and feature processing (such as fragmentation and TTL exceed). If a switchover occurs, software forwarding is disrupted until the new VSS Active supervisor engine obtains the latest CEF and other forwarding information.

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

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

The VSS achieves Layer 3 load-balancing over all paths in the FIB entries, be it local or remote.

IPv6

VSS supports IPv6 unicast and multicast as it is there on standalone system.

IPv4 Multicast

The IPv4 multicast protocols run on the VSS Active supervisor engine. Internet Group Management Protocol (IGMP) and Protocol Independent Multicast (PIM) protocol packets received on the VSS Standby supervisor engine are transmitted across VSL to the VSS Active supervisor engine. The VSS Active supervisor engine generates IGMP and PIM protocol packets to be sent over ports on either VSS member.

The VSS Active supervisor engine syncs Multicast Forwarding Information Base (MFIB) state to the VSS Standby supervisor engine. On both member switches, all multicast routes are loaded in hardware with replica expansion table (RET) entries programmed for only local outgoing interfaces. Both member switches are capable of performing hardware forwarding.

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

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

Software Features

Software features run only on the VSS Active supervisor engine. Incoming packets to the VSS Standby switch that require software processing are sent across the VSL to the VSS Active supervisor engine.

System Monitoring

The following sections describe system monitoring and system management for a VSS:

- Environmental Monitoring, page 1-20
- File System Access, page 1-20
- Diagnostics, page 1-21
- Network Management, page 1-21

Environmental Monitoring

Environmental monitoring runs on both supervisor engines. The VSS Standby switch reports notifications to the VSS Active supervisor engine. The VSS Active switch gathers log messages for both switches. The VSS Active switch synchronizes the calendar and system clock to the VSS Standby switch.

File System Access

File system access on VSS is the same as it is on dual supervisor standalone system. All files on a standby switch are accessible with slave prefix as following:

```
Switch# dir ?
  /all                  List all files
  /recursive            List files recursively
  all-filesystems       List files on all filesystems
  bootflash:            Directory or file name
  cat4000_flash:        Directory or file name
  cns:                  Directory or file name
  crashinfo:            Directory or file name
  kinfo:                Directory or file name
  null:                 Directory or file name
  nvram:                Directory or file name
  revrcsf:              Directory or file name
  slavebootflash:       Directory or file name
  slavecat4000_flash:   Directory or file name
  slavecrashinfo:       Directory or file name
  slavekinfo:           Directory or file name
  slavenvram:           Directory or file name
  slaveslot0:           Directory or file name
  slaveusb0:            Directory or file name
  slot0:                Directory or file name
  system:               Directory or file name
  tar:                  Directory or file name
  tmpsys:               Directory or file name
  usb0:                 Directory or file name
  |                     Output modifiers
```

All file or directory name with prefix "slave" show vss standby files.

Diagnostics

Bootup diagnostics are run independently on both switches. Online diagnostics can be invoked on the basis of virtual slots, which provide accessibility to modules on both switches. Use the **show switch virtual slot-map** command to display the virtual to physical slot mapping.

```
Switch# show switch virtual slot-map
Virtual Slot to Remote Switch/Physical Slot Mapping Table:

Virtual    Remote      Physical   Module
Slot No    Switch No   Slot No    Uptime
---------+-----------+----------+----------
  1          1            1          -
  2          1            2          -
  3          1            3        02:43:51
  4          1            4          -
  5          1            5          -
  6          1            6        02:45:20
  7          1            7          -
  8          1            8        02:43:50
  9          1            9          -
  10         1            10         -
  11         2            1        02:46:50
  12         2            2        02:46:50
  13         2            3          -
  14         2            4          -
  15         2            5        02:42:23
  16         2            6          -
  17         2            7          -
  18         2            8          -
  19         2            9          -
  20         2            10         -
```

Network Management

The following sections describe network management for a VSS:

- Telnet over SSH Sessions and the Web Browser User Interface, page 1-21
- SNMP, page 1-22
- Command Console, page 1-22
- Accessing the Remote Console on VSS, page 1-22
- Copying Files to Bootflash, page 1-23
- Transferring a Large File over VSL, page 1-23

Telnet over SSH Sessions and the Web Browser User Interface

A VSS supports remote access using Telnet over SSH sessions and the Cisco web browser user interface.

All remote access is directed to the VSS Active supervisor engine, which manages the whole VSS.

If the VSS performs a switchover, Telnet over SSH sessions and web browser sessions are disconnected.

SNMP

The SNMP agent runs on the VSS Active supervisor engine.

CISCO-VIRTUAL-SWITCH-MIB is a new MIB for virtual switch mode and contains the following main components:

- cvsGlobalObjects — Domain #, Switch #, Switch Mode
- cvsCoreSwitchConfig — Switch Priority
- cvsChassisTable — Switch Role and Uptime
- cvsModuleTable — Information on the physical modules listed in the ENTITY-MIB entPhysicalTable, whose entPhysicalClass is module(9)
- cvsVSLConnectionTable — VSL Port Count, Operational State
- cvsVSLStatsTable — Total Packets, Total Error Packets
- cvsVSLPortStatsTable — TX/RX Good, Bad, Bi-dir and Uni-dir Packets

Command Console

Because the management plane of the two switches are common (that is, both switches in a VSS can be configured and managed from Active switch itself), you do not require access to the Standby console. However, the consoles of both switches are available by connecting console cables to both supervisor engine console ports. Availability of the Standby console does not imply that you can configure the switch from Standby console as well. Config mode is not available on the Standby and **show** commands are limited in availability. Observe that all **show** commands, even for remote ports, are available on the Active switch.

The console on the VSS Standby switch will indicate that switch is operating in VSS Standby mode by adding the characters "-stdby" to the command line prompt. You cannot enter configuration mode on the VSS Standby switch console.

The following example shows the prompt on the VSS Standby console:

```
Switch-standby> sh clock
*14:04:58.705 UTC Tue Nov 20 2012
```

Accessing the Remote Console on VSS

Remote console (the Standby's console) can be accessed from the Local (Active) switch. This is available on a standalone system and works similarly on VSS. To access the remote console from the Active, you can use the **remote login** command with a VSS-Standby module number. Observe that the module number is a virtual slot and it would be an In-Chassis-Active supervisor module number on the remote chassis.

```
Switch# remote login module 11
Connecting to standby virtual console
Type "exit" or "quit" to end this session

      9 Switch-standby-console>
```

Because the Standby console is not available in config mode and only partially available in EXEC mode, distributed features like Netflow and Wireshark have special exemptions for respective commands (that is, these commands are allowed). Refer to Chapter 1, "Configuring Flexible NetFlow" and Chapter 1, "Configuring Wireshark" for details.

Copying Files to Bootflash

When you copy a file to a bootflash on the Active, it is not automatically copied to the Standby bootflash. This means that when you perform an ISSU upgrade or downgrade, both switches must receive the files individually. This behavior matches that on a dual-supervisor standalone system. Similarly, the removal of a file on one switch does not cause the removal of the same file on the other switch.

Transferring a Large File over VSL

Because the management plane of the VSS switches are performed through the Active, you might need to send a large-config/image file from one switch to another (that is, sending a file transfer over VSL). When you do this, the VSL link becomes "busy." Because data is flowing on a front panel port, it [the data] is significantly slower than what you might see on a dual-supervisor standalone system because in the latter, this action occurs through dedicated EOBC link.

On VSS, copying a large file from one switch to another may take several minutes. Hence, you should do this only when needed. Consider a wait of several minutes before file transfer completes.

Dual-Active Detection

If the VSL fails, the VSS Standby switch cannot determine the state of the VSS Active switch. To ensure that switchover occurs without delay, the VSS Standby switch assumes the VSS Active switch has failed and initiates switchover to take over the VSS Active role.

If the original VSS Active switch is still operational, both switch are now VSS Active. This situation is called a *dual-active scenario*. A dual-active scenario can have adverse effects on network stability, because both switches use the same IP addresses, SSH keys, and STP bridge ID. The VSS must detect a dual-active scenario and take recovery action.

The VSS supports the methods, Enhanced PAgP and Fast-Hello, for detecting a dual-active scenario. PAgP uses messaging over the MEC links to communicate between the two switches through a neighbor switch. Enhanced PAgP requires a neighbor switch that supports the PAgP enhancements.

The dual-active detection and recovery methods are described in the following sections:

- Dual-Active Detection Using Enhanced PAgP, page 1-23
- Dual-Active Detection Using Fast-Hello, page 1-24
- Recovery Actions, page 1-24

Dual-Active Detection Using Enhanced PAgP

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

In virtual switch mode, PAgP messages include a new type length value (TLV) which contains the ID of the VSS Active switch. Only switches in virtual switch mode send the new TLV.

For dual-active detection to operate successfully, one or more of the connected switches must be able to process the new TLV. Catalyst 4500, Catalyst 4500-X, and Catalyst 49*xx* series switches have this capability. For a list of other Cisco products that support enhanced PAgP, refer to Release Notes for Cisco IOS Release at this URL:

http://www.cisco.com/en/US/products/ps6350/tsd_products_support_series_home.html

When the VSS Standby switch detects VSL failure, it initiates SSO and becomes VSS Active. Subsequent PAgP messages to the connected switch from the newly VSS Active switch contain the new VSS Active ID. The connected switch sends PAgP messages with the new VSS Active ID to both VSS switches.

If the formerly VSS Active switch is still operational, it detects the dual-active scenario because the VSS Active ID in the PAgP messages changes. This switch initiates recovery actions as described in the "Recovery Actions" section on page 1-24.

Dual-Active Detection Using Fast-Hello

Dual-Active fast-hello employs fast-hello Layer 2 messages over a direct Ethernet connection. When the VSL goes down, the event is communicated to the peer switch. If the switch was operating as the active before the VSL went down, it goes into recovery mode upon receipt of a VSL down indication from the peer switch. This method is faster than IP BFD and ePAGP and does not require a neighboring switch.

Fast-Hello Link

A fast-hello link is configured between two VSS members with the intention of detecting a dual-active condition. Configuring dual-active fast-hello automatically removes all configurations from the specified interfaces, and restricts the interface to dual-active configuration commands. The following commands are allowed only in restricted mode on a fast-hello interface:

default—Sets a command to its defaults

description—Describes the interface

dual-active—Specifies a virtual switch dual-active config

exit—Exits from the fast hello interface configuration mode

load-interval—Specifies the interval for load calculation on an interface

logging—Configures logging for interface

no—Negates a command or set its defaults

shutdown—Shuts down the selected interface

No data traffic other than fast-hello can be used by fast-hello links.

For details on how to configure fast-hello dual-active detection, see the "Configuring Fast-Hello Dual-Active Detection" section on page 1-50.

Recovery Actions

An VSS Active switch that detects a dual-active condition shuts down (by err-disabling) all of its non-VSL interfaces to remove itself from the network, and waits in recovery mode until the VSL links have recovered. You might need to intervene directly to fix the VSL failure. When the shut down switch detects that VSL is operational again, the switch reloads and returns to service as the VSS Standby switch.

Loopback interfaces are also shut down in recovery mode. The loopback interfaces are operationally down and not err-disabled.

Note If the running configuration of the switch in recovery mode has been changed without saving, the switch will not automatically reload. In this situation, you must write the configuration to memory and then reload manually using the **reload** command. Only configuration changes applied to VSL ports on the switch can be saved. All other configuration changes are discarded as the node reboots as VSS standby.

When a switch becomes active (either due to dual-active scenario or otherwise), the IP address configured for fa1 management interface is associated with the active switch. By default, the switch in recovery mode will not have any IP address for the fa1 interface on its supervisor engine. To ensure IP connectivity to the switch during recovery, you ca n configure an recovery IP address. (IP address configuration is mandatory if you want IP connectivity while switch is in recovery.) When a switch enters recovery mode, the IP address for the management interface on its supervisor engine is associated with the recovery IP address.

The recovery IP address for a management interface can be verified in the output of commands such as **show ip interface brief** and **show interfaces**.

Configuring a Recovery IP Address

The recovery IP address is the IP address that is used for the fa1 interface (of a switch) while in recovery mode.

To configure the recovery IP address for the fa1 interface, perform the following task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters configuration mode.
Step 2	`Switch (config)#` **`switch virtual domain`** *`domain-id`*	Specifies virtual switch domain.
Step 3	`Switch (config-vs-domain)# [`**`no`**`]` **`dual-active recovery [switch`** *`n`***`] ip address`** *`recovery-ip-address recovery-ip-mask`*	Configures a recovery IP address. *n* is the VSS switch ID.

The following example shows how to set a recovery IP address 111.255.255.2555.0:

```
Switch# configure terminal
Switch(config)# switch virtual domain 19
Switch(config-vs-domain)# dual-active recovery ip address 1.1.1.1 255.255.255.0
```

By default, **ip address** is not configured for recovery mode. So, the switch-fa1 interface is not associated with an IP address while the switch is in recovery mode. This ensures that two devices do not respond to the same IP address.

Without the **switch** *n* option, the (same) *recovery ip address* is used by either switch when it enters recovery mode. By definition, there is only one switch (in a given VSS system) in recovery mode at a time, making one recovery ip address sufficient.

If the two switches must use different IP addresses when the respective switch is in recovery mode, use the **switch** *n* option.

You can configure recovery IP addresses without the **switch** *n* option and with the **switch** *n* option simultaneously (for a total of three IP addresses, one global and one per switch). When done, the per-switch IP address takes precedence. If no per-switch IP address exists, the global IP address is used. Following are two examples:

Scenario 1

The VSS System is configured as follows:

- Global IP address- GIP
- switch 1 IP address - IP1
- switch 2 IP address - IP2

In this scenario, if switch 1 enters recovery mode, it will use IP1 for the fa1 interface on switch 1. Conversely, if switch 2 enters recovery mode, it will use IP2 for the fa1 interface on switch2.

Scenario 2

The VSS system is configured as follows:

- Global IP address - GIP
- switch 1 IP address - IP1
- switch 2 specific IP address

In this scenario, if switch 1 enters recovery mode, it will use IP1 for the fa1 interface on the switch 1. Conversely, if switch 2 enters recovery mode, it will use GIP for the fa1 interface on switch2.

VSS Initialization

A VSS is formed when the two switches and the VSL link between them become operational. The peer switch communicates over the VSL to negotiate the switches' roles.

If only one switch becomes operational, it assumes the VSS Active role. The VSS forms when the second switch becomes operational and both switches bring up their VSL interfaces.

VSS initialization is described in the following sections:

- Virtual Switch Link Protocol, page 1-26
- SSO Dependencies, page 1-27
- Initialization Procedure, page 1-27

Virtual Switch Link Protocol

The Virtual Switch Link Protocol (VSLP) consists of several protocols that contribute to virtual switch initialization. The VSLP includes the following protocols:

- Role Resolution Protocol

  The peer switch use Role Resolution Protocol (RRP) to negotiate the role (VSS Active or VSS Standby) for each switch.

- Link Management Protocol

  The Link Management Protocol (LMP) runs on all VSL links, and exchanges information required to establish communication between the two switches.

LMP identifies and rejects any unidirectional links. If LMP flags a unidirectional link, the switch that detects the condition brings the link down and up to restart the VSLP negotiation. VSL moves the control traffic to another port if necessary.

SSO Dependencies

For the VSS to operate with SSO redundancy, the VSS must meet the following conditions:

- Identical software versions (except during ISSU with compatible versions)
- VSL configuration consistency

  During the startup sequence, the VSS Standby switch sends virtual switch information from the startup-config file to the VSS Active switch.

  The VSS Active switch ensures that the following information matches correctly on both switches:

  - Switch virtual domain
  - Switch virtual node
  - Switch priority (optional)
  - VSL port channel: switch virtual link identifier
  - VSL ports: channel-group number, shutdown, total number of VSL ports
- If the VSS detects a mismatch, it prints out an error message on the VSS Active switch console and the VSS Standby switch does not bootup. There are various ways to recover from this situation. If the switch is not running live traffic, you can either disconnect the VSL links or shutdown VSL ports on the peer, which would boot in VSS Active mode. You can make the necessary changes afterwards and reboot the switch and ensure VSL links are connected and not put in shutdown mode. Alternatively, you could clear the VSS rommon variable (VS_SWITCH_NUMBER) and allow the switch to boot in standalone mode. This method requires that no traffic flows through this switch. Once the switch is in standalone mode, you can convert it to VSS and then reboot it.
- SSO and NSF enabled

  SSO and NSF must be configured and enabled on both switches. For detailed information on configuring and verifying SSO and NSF, see Chapter 1, “Configuring Cisco NSF with SSO Supervisor Engine Redundancy.”

If these conditions are unsatisfied, the VSS stops booting and ensures that the forwarding plane is not performing forwarding. For a description of SSO and RPR, see the “VSS Redundancy” section on page 1-11.

Initialization Procedure

The following sections describe the VSS initialization procedure:

- VSL Initialization, page 1-28
- System Initialization, page 1-28
- VSL Down, page 1-28

VSL Initialization

A VSS is formed when the two switches and the VSL link between them become operational. Because both switches need to be assigned their role (VSS Active or VSS Standby) before completing initialization, VSL is brought online before the rest of the system is initialized. The initialization sequence is as follows:

1. The VSS initializes all cards with VSL ports, and then initializes the VSL ports.
2. The two switch communicate over VSL to negotiate their roles (VSS Active or VSS Standby).
3. The VSS Active switch completes the boot sequence, including the consistency check described in the "SSO Dependencies" section on page 1-27.
4. If the consistency check completed successfully, the VSS Standby switch comes up in SSO VSS Standby mode. If the consistency check failed, the VSS Standby switch comes up in RPR mode.
5. The VSS Active switch synchronizes configuration and application data to the VSS Standby switch. If VSS is either forming for the first time or a mismatch exists between VSL information sent by the Standby switch and what is on the Active switch, the new configuration is absorbed in the startup-config. This means that if the Active switch was running prior to the Standby switch and unsaved configurations existed, they would be written to the startup-config if the Standby switch sends mismatched VSL information.

System Initialization

If you boot both switches simultaneously, the switch configured as Switch 1 boots as VSS Active and the one with Switch 2 boots as VSS Standby. If priority is configured, the higher priority switch becomes active.

If you boot only one switch, the VSL ports remain inactive, and the switch boots as VSS Active. When you subsequently boot the other switch, the VSL links become active, and the new switch boots as VSS Standby. Because preemption is not supported, if a VSS Active is already running, the peer switch would always receive the VSS Standby role, even if its priority is higher than that of the Active's.

VSL Down

If the VSL is down when both switches try to boot up, the situation is similar to a dual-active scenario.

One of the switch becomes VSS Active and the other switch initiates recovery from the dual-active scenario. For further information, see the "Configuring Dual-Active Detection" section on page 1-49.

VSS Configuration Guidelines and Restrictions

The following sections describe restrictions and guidelines for VSS configuration:

- General VSS Restrictions and Guidelines, page 1-29
- Multichassis EtherChannel Restrictions and Guidelines, page 1-30
- Dual-Active Detection Restrictions and Guidelines, page 1-30

General VSS Restrictions and Guidelines

When configuring the VSS, note the following guidelines and restrictions:

- In Cisco IOS XE 3.4.0E (15.1(2)SG, E, VSS did not support SMI (both Director and Client).

  Beginning with Cisco IOS XE 3.5.0E (15.2(1)E, VSS supports SmartInstall Director but not SMI Client.

  VSS [mode] is transparent to SMI except for the changes in interface names.
- The SMI Director has only one instance on VSS and runs on the VSS active switch. The standby Catalyst 4500 switch in a VSS is not listed as a director in the output of the **show vstack status** command. In order to list both the VSS active and standby as a directors, enter the **redundancy force-switchover** command. After the current VSS standby takes over as the VSS active, the **show vstack status** command lists two directors.
- The VSS configurations in the startup-config file must match on both switches; that is, the domain must match, the switch ID must be unique, and the VSL ports' information must match the physical connection.
- There is no restriction to configure oversubscribed linecard ports as VSL. The responsibility of bandwidth availability for a given network requirement lies with the network operator.
- VSL portchannel must have more than one port in the channel, preferably distributed on more than one module. If the VSL consists of only one link, its failure causes a Dual-Active operation of the VSS. Also, all VSL links configured on one module may cause a Dual-Active operation, if the module goes down..
- The ICS supervisor engine is supported only in rommon mode; its ports are available but the supervisor engine neither forwards traffic nor provides any redundancy in that chassis.
- If a dual-supervisor system is being converted to VSS, each supervisor engine in the chassis must be converted to VSS one at a time; when one supervisor is being converted to VSS, another one must remain in rommon or be removed from the chassis. When both supervisor engines are converted, they could be inserted in the chassis. A combination of converted and non-converted supervisor engines in a chassis is not supported and it may disrupt the network.
- Classification and marking based on 'qos-group' in a QoS policy-map is not supported in VSS.
- The following older gneration linecards (WS-X42xy to WS-X45xy) are supported with the VSS feature:
  - WS-X4148-RJ
  - WS-X4148-RJ
  - WS-X4148-FX-MT
  - WS-X4306-GB
  - WS-X4548-RJ45V+
  - WS-X4448-GB-SFP
  - WS-X4248-FE-SFP
  - WS-X4248-RJ45V

  Please remove all other linecards from your system when converting from standalone to VSS mode.

- Do not attach a QoS policy with the maximum queue-limit (8184) to a large number of targets in a VSS system. This will cause continuous reloads on the standby supervisor engine.
- When an aymmetric virtual switch (i.e. a VSS comprising of chassis with different slot capacities) boots initially after conversion from standalone mode, the entPhysicalDescr object for the standby chassis does not hold the correct value. The entPhysicalDescr objects for both the active and standby chassis will match and hold the value for the active chassis.

  After the running configuration is saved and a shelf reload occurs, this behaviour is not observed - the entPhysicalDescr objects for both chassis accurately reflects the correct chassis types.

Multichassis EtherChannel Restrictions and Guidelines

When configuring MECs, note the following guidelines and restrictions:

- Port Security over EtherChannels is not supported.
- All links in an MEC must terminate locally on the VSS Active or VSS Standby switch of the same virtual domain.
- An MEC can be connected to another MEC on a different VSS domain.
- Policers applied on an MEC are applied on two switches independently; if a policer is applied for 100 Mbps of conforming action, it will apply 100Mbps on both switches, resulting in a total conforming rate of 200 Mbps. To mitigate this, you can reduce the policer rate. In a more restrictive case, a rate of 50 Mbps might be necessary to achieve a maximum of 100Mbps. In a more liberal case, where conforming action of 200 Mbps is not a problem, policing rate could be kept to 100Mbps.

Dual-Active Detection Restrictions and Guidelines

When configuring dual-active detection, note the following guidelines and restrictions:

- For line redundancy, we recommend configuring at least two ports per switch for dual-active detection. For module redundancy, the two ports can be on different modules in each switch, and should be on different modules than the VSL ports, if feasible.
- Only trusted PAgP channels are relied upon to detect dual-active mode of operation.

Configuring a VSS

These sections describe how to configure a VSS:

- Converting to a VSS, page 1-31
- Displaying VSS Information, page 1-36
- Converting a VSS to Standalone Switch, page 1-37
- Configuring VSS Parameters, page 1-39
- Configuring Multichassis EtherChannels, page 1-45
- Configuring Dual-Active Detection, page 1-49

Converting to a VSS

By default, the Catalyst 4500/4500X series switch is configured to operate in standalone mode (the switch works independently). The VSS combines two standalone switches into one virtual switch, operating in virtual switch mode.

Note When you convert two standalone switches into one VSS, all non-VSL configuration settings on the VSS Standby switch will revert to the default configuration.

Note Preferably, conversion to VSS should be done on a maintenance window. If you plan to use the same port channel number for VSL, default the existing port channel configurations that are available on standalone switches. Then, follow the guidelines in section Configuring VSL Port Channel and Ports, page 1-33.

To convert two standalone switches into a VSS, you perform the following major activities:

- Save the standalone configuration files.
- Configure each switch for required VSS configurations.
- Convert to a VSS.

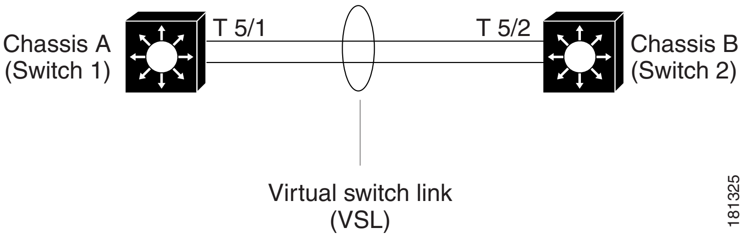
In virtual switch mode, both switches use the same configuration file. When you make configuration changes on the VSS Active switch, these changes are automatically propagated to the VSS Standby switch.

The tasks required to convert the standalone switch to a VSS are detailed in the following sections:

- Backing Up the Standalone Configuration, page 1-32
- Configuring SSO and NSF, page 1-32
- Assigning Virtual Switch Domain and Switch Numbers, page 1-32
- Configuring VSL Port Channel and Ports, page 1-33
- Converting the Switch to Virtual Switch Mode, page 1-34
- (Optional) Configuring VSS Standby Switch Modules, page 1-35

In the procedures that follow, the example commands assume the configuration shown in Figure 1-8.

Figure 1-8 Example VSS



Two chassis, A and B, are converted into a VSS with virtual switch domain 100. Interface 10-Gigabit Ethernet 5/1 on Switch 1 is connected to interface 10-Gigabit Ethernet 5/2 on Switch 2 to form the VSL.

Note The port channels 10 and 20 mentioned in the config steps below are merely exemplary. You can configure any port channel number from 1-64 for VSL port channel.

Backing Up the Standalone Configuration

Save the configuration files for both switches operating in standalone mode. You need these files to revert to standalone mode from virtual switch mode.

On Switch 1, perform this task:

	Command	Purpose
Step 1	Switch-1# **copy running-config startup-config**	(Optional) Saves the running configuration to startup configuration.
Step 2	Switch-1# **copy startup-config disk0:old-startup-config**	Copies the startup configuration to a backup file.

On Switch 2, perform this task:

	Command	Purpose
Step 1	Switch-2# **copy running-config startup-config**	(Optional) Saves the running configuration to the startup configuration file.
Step 2	Switch-2# **copy startup-config disk0:old-startup-config**	Copies the startup configuration to a backup file.

Configuring SSO and NSF

SSO and NSF are configured as default on VSS.

Assigning Virtual Switch Domain and Switch Numbers

You must configure the same virtual switch domain number on both switches of the VSS. The virtual switch domain is a number between 1 and 255, and must be unique for each VSS in your network (the domain number is incorporated into various identifiers to ensure that these identifiers are unique across the network).

Within the VSS, you must configure one switch to be switch number 1 and the other switch to be switch number 2.

To configure the virtual switch domain and switch number on both switches, perform this task on Switch 1:

	Command	Purpose
Step 1	Switch-1(config)# **switch virtual domain 100**	Configures the virtual switch domain on Switch A.
Step 2	Switch-1(config-vs-domain)# **switch 1**	Configures Switch A as virtual switch number 1.
Step 3	Switch-1(config-vs-domain)# **exit**	Exits config-vs-domain.

Perform the following task on Switch 2:

	Command	Purpose
Step 1	`Switch-2(config)# switch virtual domain 100`	Configures the virtual switch domain on Switch B.
Step 2	`Switch-2(config-vs-domain)# switch 2`	Configures Switch B as virtual switch number 2.
Step 3	`Switch-2(config-vs-domain)# exit`	Exits config-vs-domain.

Note The switch number is not stored in the startup or running configuration, because both switches use the same configuration file (but must not have the same switch number).

Configuring VSL Port Channel and Ports

The VSL is configured with a unique port channel on each switch. During the conversion, the VSS configures both port channels on the VSS Active switch. If the VSS Standby switch VSL port channel number has been configured for another use, the VSS comes up in RPR mode. To avoid this situation, check that both port channel numbers are available on both of the switches.

Check the port channel number with the **show running-config interface port-channel** command. The command displays an error message if the port channel is available for VSL. For example, the following command shows that port channel 20 is available on Switch 1:

```
Switch-1 # show running-config interface port-channel 20
% Invalid input detected at '^' marker.
```

To configure the VSL port channels, perform this task on Switch 1:

Note The port channels 10 and 20 mentioned in the configuration steps below are exemplary only. You can configure any port channel number from 1-64 for VSL port channel.

	Command	Purpose
Step 1	`Switch-1(config)# interface port-channel 10`	Configures port channel 10 on Switch 1.
Step 2	`Switch-1(config)# switchport`	Convert to a Layer 2 port.
Step 3	`Switch-1(config-if)# switch virtual link 1`	Associates Switch 1 as owner of port channel 10.
Step 4	`Switch-1(config-if)# no shutdown`	Activates the port channel.
Step 5	`Switch-1(config-if)# exit`	Exits interface configuration.

Perform the following task on Switch 2:

	Command	Purpose
Step 1	`Switch-2(config)# interface port-channel 20`	Configures port channel 20 on Switch 2.
Step 2	`Switch-1(config)# switchport`	Convert to a Layer 2 port.
Step 3	`Switch-2(config-if)# switch virtual link 2`	Associates Switch 2 as owner of port channel 20.

	Command	Purpose
Step 4	`Switch-2(config-if)# no shutdown`	Activates the port channel.
Step 5	`Switch-2(config-if)# exit`	Exits interface configuration mode.

You must add the VSL physical ports to the port channel. In the following example, interfaces 10-Gigabit Ethernet 3/1 and 3/2 on Switch 1 are connected to interfaces 10-Gigabit Ethernet 5/2 and 5/3 on Switch 2.

Tip For line redundancy, we recommend configuring at least two ports per switch for the VSL. For module redundancy, the two ports can be on different switching modules in each chassis.

To configure the VSL ports, perform this task on Switch 1:

	Command	Purpose
Step 1	`Switch-1(config)# interface range tengigabitethernet 3/1-2`	Enters configuration mode for interface range tengigabitethernet 3/1-2 on Switch 1.
Step 2	`Switch-1(config-if)# channel-group 10 mode on`	Adds this interface to channel group 10.

Note 1G ports, which are converted from 10G ports using a connector, are not supported for VSL. This impacts Sup7-E and Sup7L-E ports.

On Switch 2, perform this task:

	Command	Purpose
Step 1	`Switch-2(config)# interface range tengigabitethernet 5/2-3`	Enters configuration mode for interface range tengigabitethernet 5/2-3 on Switch 2.
Step 2	`Switch-2(config-if)# channel-group 20 mode on`	Adds this interface to channel group 20.

Note 1G ports, which are converted from 10G ports using a connector, are not supported for VSL. This impacts Sup7-E and Sup7L-E ports.

Converting the Switch to Virtual Switch Mode

Conversion to virtual switch mode requires a restart for both switches. After the reboot, commands that specify interfaces with module/port now include the switch number. For example, a port on a switching module is specified by switch/module/port.

Prior to the restart, the VSS converts the startup configuration to use the switch/module/port convention. A backup copy of the startup configuration file is saved in bootflash. This file is assigned a default name, but you are also prompted to override the default name if you want to change it.

To convert Switch 1 to virtual switch mode, perform this task:

Command	Purpose
`Switch-1# switch convert mode virtual`	Converts Switch 1 to virtual switch mode. After you enter the command, you are prompted to confirm the action. Enter **yes**. The system creates a converted configuration file, and saves the file to the bootflash.

To convert Switch 2 to virtual switch mode, perform this task on Switch 2:

Command	Purpose
`Switch-2# switch convert mode virtual`	Converts Switch 2 to virtual switch mode. After you enter the command, you are prompted to confirm the action. Enter **yes**. The system creates a converted configuration file, and saves the file to the bootflash.

Note After you confirm the command (by entering **yes** at the prompt), the running configuration is automatically saved as the startup configuration and the switch reboots. After the reboot, the switch is in virtual switch mode, so you must specify interfaces with three identifiers (switch/module/port).

When switches are being converted to VSS, you should not set them to ignore startup-config. If done, the switch can be enabled to parse the startup-config at the rommon prompt. Ignoring startup-config in VSS mode, causes a switch to boot in a semi-VSS mode, which can only be corrected by a reboot and by enabling the parsing of startup-config.

(Optional) Configuring VSS Standby Switch Modules

Note You cannot configure or provision modules on VSS.

When switches form initial VSS relationships, they send module information to each other and this information is pushed to the configuration and used subsequently for provisioning, provided the switch is booting and the peer is down or not present.

The following example shows the module provisioning information:

```
module provision switch 1
 slot 1 slot-type 148 port-type 60 number 4  virtual-slot 17
 slot 2 slot-type 137 port-type 31 number 16  virtual-slot 18
 slot 3 slot-type 227 port-type 60 number 8  virtual-slot 19
 slot 4 slot-type 225 port-type 61 number 48  virtual-slot 20
 slot 5 slot-type 82 port-type 31 number 2  virtual-slot 21
module provision switch 2
 slot 1 slot-type 148 port-type 60 number 4  virtual-slot 33
 slot 2 slot-type 227 port-type 60 number 8  virtual-slot 34
 slot 3 slot-type 137 port-type 31 number 16  virtual-slot 35
 slot 4 slot-type 225 port-type 61 number 48  virtual-slot 36
 slot 5 slot-type 82 port-type 31 number 2  virtual-slot 37
```

These commands are not available to the user and that various numbers used in these commands are internal to the system and used to identify a module. These commands are written to the startup-config when a switch detects a given module while it is running in VSS mode. When reconverted to standalone mode, these commands are removed from the startup-config.

Displaying VSS Information

To display basic information about the VSS, perform one of these tasks:

Command	Purpose
Switch# **show switch virtual**	Displays the virtual switch domain number, and the switch number and role for each of the switches.
Switch# **show switch virtual role**	Displays the role, switch number, and priority for each of the switch in the VSS.
Switch# **show switch virtual link**	Displays the status of the VSL.

The following example shows the information output from these commands:

```
Switch# show switch virtual
Executing the command on VSS member switch role = VSS Active, id = 1

Switch mode                  : Virtual Switch
Virtual switch domain number : 100
Local switch number          : 1
Local switch operational role: Virtual Switch Active
Peer switch number           : 2
Peer switch operational role : Virtual Switch Standby

Executing the command on VSS member switch role = VSS Standby, id = 2

Switch mode                  : Virtual Switch
Virtual switch domain number : 100
Local switch number          : 2
Local switch operational role: Virtual Switch Standby
Peer switch number           : 1
Peer switch operational role : Virtual Switch Active

Switch# show switch virtual role

Executing the command on VSS member switch role = VSS Active, id = 1

RRP information for Instance 1

--------------------------------------------------------------------
Valid  Flags   Peer      Preferred  Reserved
               Count     Peer       Peer

--------------------------------------------------------------------
TRUE    V        1           1          1

Switch  Switch Status  Preempt       Priority  Role      Local    Remote
        Number         Oper(Conf)    Oper(Conf)          SID      SID
--------------------------------------------------------------------
```

```
LOCAL   1      UP      FALSE(N )     100(100)  ACTIVE    0       0
REMOTE  2      UP      FALSE(N )     100(100)  STANDBY   7496    7678

Peer 0 represents the local switch

Flags : V - Valid
In dual-active recovery mode: No


Executing the command on VSS member switch role = VSS Standby, id = 2

RRP information for Instance 2

--------------------------------------------------------------------
Valid  Flags    Peer      Preferred  Reserved
                Count     Peer       Peer

-------------------------------------------------------------------
TRUE    V        1           1          1

Switch  Switch Status  Preempt       Priority  Role      Local    Remote
Number         Oper(Conf)    Oper(Conf)         SID      SID

--------------------------------------------------------------------
LOCAL   2      UP      FALSE(N )     100(100)  STANDBY   0       0
REMOTE  1      UP      FALSE(N )     100(100)  ACTIVE    7678    7496

Peer 0 represents the local switch

Flags : V - Valid
In dual-active recovery mode: No

Switch# show switch virtual link

Executing the command on VSS member switch role = VSS Active, id = 1


VSL Status : UP
VSL Uptime : 13 minutes
VSL Control Link : Te1/1/1

Executing the command on VSS member switch role = VSS Standby, id = 2


VSL Status : UP
VSL Uptime : 13 minutes
VSL Control Link : Te2/1/1
```

Converting a VSS to Standalone Switch

To convert a VSS into two standalone systems, you perform the following major steps:

- Copying the VSS Configuration to a Backup File, page 1-38
- Converting the VSS Active Switch to Standalone, page 1-38
- Converting the VSS Standby Switch to Standalone, page 1-38

Copying the VSS Configuration to a Backup File

Save the configuration file from the VSS Active switch. You may need this file if you convert to virtual switch mode again. You only need to save the file from the VSS Active switch, because the configuration file on the VSS Standby switch is identical to the file on the VSS Active switch.

	Command	Purpose
Step 1	`Switch-1# copy running-config startup-config`	(Optional) Saves the running configuration to startup configuration. This step is only required if there are unsaved changes in the running configuration that you want to preserve.
Step 2	`Switch-1# copy startup-config bootflash:vs-startup-config`	Copies the startup configuration to a backup file.

Converting the VSS Active Switch to Standalone

When you convert the VSS Active switch to standalone mode, the VSS Active switch removes the provisioning and configuration information related to VSL links and the peer chassis modules, saves the configuration file, and performs a reload. The switch comes up in standalone mode with only the configuration data relevant to the standalone system.

The VSS Standby switch of the VSS becomes VSS Active. VSL links on this switch are down because the peer is now unavailable.

To convert the VSS Active switch to standalone mode, perform this task on the VSS Active switch:

Command	Purpose
`Switch-1# switch convert mode stand-alone`	Converts Switch 1 to standalone mode. After you enter the command, you are prompted to confirm the action. Enter **yes**.

Conversion from VSS to standalone causes all physical interfaces to be administratively shutdown and written to the startup-config. This is a safeguard against a standalone system arriving in the network alive and conflicting with a bridge or router MAC address, which might still be there if one of the VSS switches is still running in VSS mode.

We do not recommend that you convert a VSS to standalone in a live network.

Converting the VSS Standby Switch to Standalone

When you convert the new VSS Active switch to standalone mode, the switch removes the provisioning and configuration information related to VSL links and the peer switch modules, saves the configuration file and performs a reload. The switch comes up in standalone mode with only its own provisioning and configuration data.

To convert the peer switch to standalone, perform this task on the VSS Standby switch:

Command	Purpose
`Switch-2# switch convert mode stand-alone`	Converts Switch 2 to standalone mode. After you enter the command, you are prompted to confirm the action. Enter **yes**.

Configuring VSS Parameters

These sections describe how to configure VSS parameters:

- Configuring VSL Switch Priority, page 1-39
- Configuring a VSL, page 1-41
- Adding and Deleting a VSL Port After the Bootup, page 1-41
- Displaying VSL Information, page 1-42
- Configuring VSL QoS, page 1-43
- Configuring the Router MAC Address, page 1-44

Configuring VSL Switch Priority

To configure the switch priority, perform this task:

	Command	Purpose
Step 1	`Switch(config)# switch virtual domain 100`	Enters configuration mode for the virtual switch domain.

	Command	Purpose
Step 2	`Switch(config-vs-domain)# switch [1 | 2] priority [priority_num]`	Configures the priority for the switch. The switch with the higher priority assumes the VSS Active role. The range is 1 (lowest priority) to 255 (highest priority); the default is 100. **Note** • The new priority value only takes effect after you save the configuration and perform a reload of the VSS. • If the higher priority switch is currently in VSS Standby state, you can make it the VSS Active switch by initiating a switchover with the **redundancy force-switchover** command. The **show switch virtual role** command displays the operating priority and the configured priority for each switch in the VSS. • The **no** form of the command resets the priority value to the default value of 100. The new value takes effect after you save the configuration and perform a reload.
Step 3	`Switch# show switch virtual role`	Displays the current priority.

Note If you make configuration changes to the switch priority, the changes only take effect after you save the running configuration to the startup configuration file and perform a reload. The **show switch virtual role** command shows the operating and configured priority values. You can manually set the VSS Standby switch to VSS Active using the **redundancy force-switchover** command.

This example shows how to configure virtual switch priority:

```
Switch(config)# switch virtual domain 100
Switch(config-vs-domain)# switch 1 priority 200
Switch(config-vs-domain)# exit
```

This example shows how to display priority information for the VSS:

```
Switch# show switch virtual role
Switch  Switch Status  Preempt    Priority  Role     Session ID
        Number         Oper(Conf) Oper(Conf)         Local  Remote
------------------------------------------------------------------
LOCAL    1      UP      FALSE(N)   100(200)  ACTIVE   0      0
REMOTE   2      UP      FALSE(N)   100(100)  STANDBY  8158   1991

In dual-active recovery mode: No
```

Configuring a VSL

To configure a port channel to be a VSL, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface port-channel channel_num`	Enters configuration mode for the specified port channel.
Step 2	`Switch(config-if)# switch virtual link switch_num`	Assigns the port channel to the virtual link for the specified switch.

Note We recommend that you configure the VSL prior to converting the switch into a VSS.

This example shows how to configure the VSL:

```
Switch-1(config)# interface port-channel 10
Switch-1(config-if)# switch virtual link 1
Switch-1(config-if)# no shutdown (If the port is admin shutdown)
Switch-1(config)# interface tenGigabitEthernet 5/1
Switch-1(config-if)# channel-group 10 mode on
Switch-1(config-if)# no shutdown (If the port is admin shutdown)

Switch-2(config)# interface port-channel 25
Switch-2(config-if)# switch virtual link 2
Switch-2(config-if)# no shutdown (If the port is admin shutdown)
Switch-2(config-if)# interface tenGigabitEthernet 5/2
Switch-2(config-if)# channel-group 25 mode on
Switch-2(config-if)# no shutdown (If the port is admin shutdown)
```

Adding and Deleting a VSL Port After the Bootup

At any time, you can add and delete VSL ports from a port-channel to increase the nunber of links in the VSL, to move the port from one port to another, or to remove it from VSL.

Before adding or deleting VSL ports, do the following:

- Ensure all ports are physically connected to the peer switch. The peer port must also be configured for VSL.
- Shutdown the port before configuring VSL. When both ports on the link are configured for VSL, **unshut** them.
- Spread VSL ports across multiple modules.
- While deleting a port, retain at least one “active” VSL port pair. Else, a dual-active operation could occur.
- To save link flap and high CPU, shutdown the ports before VSL is unconfigured.
- After adding, deleting, or modifying VSL ports, write the config to nvram (that is, startup-config).
- If you need to move ports to another port, account for the bandwidth requirement of VSL. You should add an additional VSL link in the channel, move ports and remove additional links in the channel.

Displaying VSL Information

To display information about the VSL, perform one of these tasks:

Command	Purpose
`Switch#` **`show switch virtual link`**	Displays information about the VSL.
`Switch#` **`show switch virtual link port-channel`**	Displays information about the VSL port channel.
`Switch#` **`show switch virtual link port`**	Displays information about the VSL ports.

This example shows how to display VSL information:

```
Switch# show switch virtual link
VSL Status : UP
VSL Uptime : 1 day, 3 hours, 39 minutes
VSL Control Link : Te 1/5/1

Switch# show switch virtual link port-channel

Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      N - not in use, no aggregation
        f - failed to allocate aggregator

        M - not in use, no aggregation due to minimum links not met
        m - not in use, port not aggregated due to minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated

Group  Port-channel  Protocol    Ports
------+-------------+-----------+---------------------------------------------
10     Po10(RU)         -        Te1/5/4(P) Te1/5/5(P)
20     Po20(RU)         -        Te2/5/4(P) Te2/5/5(P)

Switch# show switch virtual link port
LMP summary

    Link info:        Configured: 1        Operational: 1

                            Peer Peer           Peer   Peer      Timer(s)running
Interface Flag State        Flag MAC            Switch Interface (Time remaining)
--------------------------------------------------------------------------------
Gi1/3/11  vfsp operational  vfsp f866.f296.be00 2      Gi2/1/11  T4(708ms)
                                                                 T5(29.91s)

 Flags:  v - Valid flag set        f - Bi-directional flag set
         s - Negotiation flag set  p - Peer detected flag set

 Timers: T4 - Hello Tx Timer   T5 - Hello Rx Timer


  LMP Status

          Last operational        Current packet          Last Diag   Time since
Interface Failure state           State                   Result      Last Diag
-------------------------------------------------------------------------------
Gi1/3/11  No failure              Hello bidir             Never ran   --
```

```
  LMP hello timer

                         Hello Tx (T4) ms          Hello Rx (T5*) ms
Interface   State        Cfg     Cur     Rem       Cfg     Cur     Rem
-------------------------------------------------------------------------
Gi1/3/11    operational  -       1000    708       -       30000   29144

*T5 = min_rx * multiplier
 Cfg : Configured Time
 Cur : Current Time
 Rem : Remaining Time
```

Configuring VSL QoS

When a physical port is configured as a member of a VSL port-channel, a queuing policy is automatically attached to the VSL member ports. This queuing policy provides a dedicated queue for VSS Management, VSLP, BFD, Layer 2 and Layer 3 control protocols, and voice and video data traffic. Each queue is provided with a minimum bandwidth, ensuring that VSS management and control protocol packets are not dropped when congestion occurs on the VSL. The bandwidth assigned to a class of traffic is the minimum bandwidth that is guaranteed to the class during congestion. The VSL link uses Transmit Queue Sharing, where the output link bandwidth is shared among multiple queues of a given VSL port. Any modification or removal of VSL Queuing policy is restricted in a VSS system.

The following command sequence is inserted automatically by software.

```
interface TenGigabitEthernet1/1/1
 switchport mode trunk
 switchport nonegotiate
 no lldp transmit
 no lldp receive
 no cdp enable
 channel-group 10 mode on
 service-policy output VSL-Queuing-Policy
end

Switch# show policy-map VSL-Queuing-Policy
  Policy Map VSL-Queuing-Policy
    Class VSL-MGMT-PACKETS
      bandwidth percent 5
    Class VSL-L2-CONTROL-PACKETS
      bandwidth percent 5
    Class VSL-L3-CONTROL-PACKETS
      bandwidth percent 5
    Class VSL-VOICE-VIDEO-TRAFFIC
      bandwidth percent 30
    Class VSL-SIGNALING-NETWORK-MGMT
      bandwidth percent 10
    Class VSL-MULTIMEDIA-TRAFFIC
      bandwidth percent 20
    Class VSL-DATA-PACKETS
      bandwidth percent 20
    Class class-default
      bandwidth percent 5

class-map match-any VSL-MGMT-PACKETS
  match access-group name VSL-MGMT

class-map match-any VSL-DATA-PACKETS
  match any

class-map match-any VSL-L2-CONTROL-PACKETS
```

```
  match access-group name VSL-DOT1x
  match access-group name VSL-BPDU
  match access-group name VSL-CDP
  match access-group name VSL-LLDP
  match access-group name VSL-SSTP
  match access-group name VSL-GARP

class-map match-any VSL-L3-CONTROL-PACKETS
  match access-group name VSL-IPV4-ROUTING
  match access-group name VSL-BFD
  match access-group name VSL-DHCP-CLIENT-TO-SERVER
  match access-group name VSL-DHCP-SERVER-TO-CLIENT
  match access-group name VSL-DHCP-SERVER-TO-SERVER
  match access-group name VSL-IPV6-ROUTING

class-map match-any VSL-MULTIMEDIA-TRAFFIC
  match  dscp af41
  match  dscp af42
  match  dscp af43
  match  dscp af31
  match  dscp af32
  match  dscp af33
  match  dscp af21
  match  dscp af22
  match  dscp af23

class-map match-any VSL-VOICE-VIDEO-TRAFFIC
  match  dscp ef
  match  dscp cs4
  match  dscp cs5

class-map match-any VSL-SIGNALING-NETWORK-MGMT
  match  dscp cs2
  match  dscp cs3
  match  dscp cs6
  match  dscp cs7
```

Configuring the Router MAC Address

On VSS, all routing protocols are centralized on the active supervisor engine. A common router MAC address is used for Layer 3 interfaces on both active and standby switches. Additionally, to ensure non-stop forwarding, the same router MAC address is used after switchover to Standby, so that all layer 3 peers see a consistent router MAC address.

There are three ways to configure a router MAC address on VSS:

- HHH—Manually set a router MAC address. Ensure that this MAC address is reserved for this usage.
- chassis—Use the mac-address range reserved for Chassis. This is the Cisco MAC address assigned to the chassis.
- use-virtual—Use the mac-address range reserved for the VSS. This is the served Cisco MAC address pool, which is derived from a base MAC address +vvs domain-id.

By default, the virtual domain based router MAC address is used. Any change of router MAC address configuration requires a reboot of both VSS supervisor engines

The follow table shows how to configure the router MAC address.

Command	Purpose
`Switch(config)# switch virtual domain domain_id`	Enters VSS configuration mode.
`Switch(config-vs-domain)# mac-address use-virtual`	Assigns the router MAC address from a reserved pool of domain-based addresses. **Note** This is the default. This is shown in the configuration, even if it the default.
`Switch(config-vs-domain)# mac-address mac-address`	Assigns the router MAC address in three 2-byte hexadecimal numbers.
`Switch(config-vs-domain)# mac-address chassis`	Specifies the router MAC address as the last address of chassis MAC address range.

Configuring Multichassis EtherChannels

Configure multichassis EtherChannels (MECs) as you would for a regular EtherChannel. The VSS will recognize that the EtherChannel is an MEC when ports from both switches are added to the EtherChannel. You can verify the MEC configuration by entering the **show etherchannel** command.

One VSS supports a maximum of 256 port channels.

To configure Layer 3 Multichassis EtherChannels, create the port channel logical interface and then put the Ethernet interfaces from both the VSS active and VSS standby into the port channel.

These sections describe Layer 3 EtherChannel configuration:

- Creating Port Channel Logical Interfaces, page 1-45
- Configuring Physical Interfaces as Layer 3 EtherChannels, page 1-46

Creating Port Channel Logical Interfaces

Note To move an IP address from a physical interface to an EtherChannel, you must delete the IP address from the physical interface before configuring it on the port channel interface.

To create a port channel interface for a Layer 3 EtherChannel, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface port-channel port_channel_number`	Creates the port channel interface. The value for *port_channel_number* can range from 1 to 64.
Step 2	`Switch(config-if)# ip address ip_address mask`	Assigns an IP address and subnet mask to the EtherChannel.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show running-config interface port-channel port_channel_number`	Verifies the configuration.

This example shows how to create port channel interface 1:

```
Switch# configure terminal
Switch(config)# interface port-channel 1
Switch(config-if)# ip address 172.32.52.10 255.255.255.0
Switch(config-if)# end
```

This example shows how to verify the configuration of port channel interface 1:

```
Switch# show running-config interface port-channel 1
Building configuration...

Current configuration:
!
interface Port-channel1
 ip address 172.32.52.10 255.255.255.0
end

Switch#
```

Configuring Physical Interfaces as Layer 3 EtherChannels

To configure physical interfaces as Layer 3 EtherChannels, perform this task for each interface:

	Command	Purpose
Step 1	Switch(config)# **interface** {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port*	Selects a physical interface to configure.
Step 2	Switch(config-if)# **no switchport**	Makes this a Layer 3 routed port.
Step 3	Switch(config-if)# **no ip address**	Ensures that no IP address is assigned to the physical interface.
Step 4	Switch(config-if)# **channel-group** *port_channel_number* **mode** {**active** | **on** | **auto** | **passive** | **desirable**}	Configures the interface in a port channel and specifies the PAgP or LACP mode. If you use PAgP, enter the keywords **auto** or **desirable**. If you use LACP, enter the keywords **active** or **passive**.
Step 5	Switch(config-if)# **end**	Exits configuration mode.
Step 6	Switch# **show running-config interface port-channel** *port_channel_number* Switch# **show running-config interface** {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* Switch# **show interfaces** {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* **etherchannel** Switch# **show etherchannel 1 port-channel**	Verifies the configuration.

This example shows how to configure Gigabit Ethernet interfaces 1/3/26 and 2/2/26 into port channel 1 with PAgP mode **desirable**:

```
Switch(config)# conf terminal
Switch(config)# int gigabitEthernet 1/3/26
Switch(config-if)# no switchport
Switch(config-if)# no ip address
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# int gigabitEthernet 2/2/6
Switch(config-if)# no switchport
Switch(config-if)# no ip address
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# end
```

Note See the "Configuring a Range of Interfaces" section on page 1-4 for information about the **range** keyword.

The following two examples show how to verify the configuration of GigabitEthernet interface 1/3/26:

```
Switch# show running-config interface gigabitEthernet 1/3/26
Building configuration...

Current configuration : 101 bytes
!
interface GigabitEthernet1/3/26
 no switchport
 no ip address
 channel-group 1 mode desirable
end

Switch# show interfaces gigabitEthernet 1/3/26 etherchannel
Port state    = Up Mstr In-Bndl
Channel group = 1           Mode = Desirable-Sl    Gcchange = 0
Port-channel  = Po1         GC   = 0x00010001      Pseudo port-channel = Po1
Port index    = 0           Load = 0x00            Protocol =   PAgP

Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.        P - Device learns on physical port.
        d - PAgP is down.
Timers: H - Hello timer is running.        Q - Quit timer is running.
        S - Switching timer is running.    I - Interface timer is running.

Local information:
                                Hello    Partner  PAgP     Learning  Group
Port      Flags State   Timers  Interval Count   Priority   Method  Ifindex
Gi1/3/26  SC    U6/S7   H       30s      1        128        Any      632

Partner's information:

          Partner              Partner          Partner         Partner Group
Port      Name                 Device ID        Port       Age  Flags   Cap.
Gi1/3/26  3750x                2c54.2dd4.ad80   Gi1/0/25    20s SAC     50001

Age of the port in the current state: 0d:00h:04m:04s
Switch#
```

This example shows how to verify the configuration of port channel interface 1 after the interfaces have been configured:

```
Switch# show etherchannel 1 port-channel
                Port-channels in the group:
                ---------------------------

Port-channel: Po1
------------

Age of the Port-channel   = 0d:00h:53m:41s
Logical slot/port   = 21/1          Number of ports = 2
GC                  = 0x00010001
Passive port list   = Gi1/3/26 Gi2/2/26
Port state          = Port-channel L3-Ag Ag-Inuse
Protocol            =   PAgP
```

```
Port security       = Disabled

Ports in the Port-channel:

Index   Load   Port     EC state        No of bits
------+------+------+------------------+-----------
  0     00     Gi1/3/26 Desirable-Sl       0
  1     00     Gi2/2/26 Desirable-Sl       0

Time since last port bundled:    0d:00h:05m:25s    Gi2/2/26
 Switch#
```

This example shows how to display a one-line summary per channel group:

```
Switch# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 3
Number of aggregators:           3

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(RU)         PAgP      Gi1/3/26(P) Gi2/2/26(P)
10     Po10(SU)         -        Te1/1/1(P)  Te1/1/4(D)
20     Po20(SU)         -        Te2/1/1(P)
```

Prior to Cisco Release IOS XE 3.5.0E and IOS 15.2(1)E, when you tried to add a port to an EtherChannel from different chassis of the VSS system, an error message displayed:

```
Switch(config)# int gi2/3/26
Switch(config-if)# no switchport
Switch(config-if)# channel-group 50 mode on
Switch(config-if)#
Switch(config)# int gi1/1/48
Switch(config-if)# no switchport
Switch(config-if)# channel-group 50 mode on
Layer 3 MEC is not supported: GigabitEthernet1/1/48 on switch 1 cannot be part of
port-channel 50 with members on switch 2.
Command rejected: conflicts with Unsupported Layer 3 MEC
Switch(config-if)#
```

Configuring Dual-Active Detection

The following sections describe how to configure dual-active detection:

- Configuring Enhanced PAgP Dual-Active Detection, page 1-49
- Configuring Fast-Hello Dual-Active Detection, page 1-50
- Displaying Dual-Active Detection, page 1-51

Configuring Enhanced PAgP Dual-Active Detection

If enhanced PAgP is running on the MECs between the VSS and its access switches, the VSS can use enhanced PAgP messaging to detect a dual-active scenario.

By default, PAgP dual-active detection is enabled. However, the enhanced messages are only sent on port channels with trust mode enabled (see the trust mode description in the note).

Note Before changing PAgP dual-active detection configuration, ensure that all port channels with trust mode enabled are in administrative down state. Use the **shutdown** command in interface configuration mode for the port channel. Remember to use the **no shutdown** command to reactivate the port channel when you are finished configuring dual-active detection.

To enable or disable PAgP dual-active detection, perform this task:

	Command	Purpose
Step 1	`Switch(config)# switch virtual domain domain_id`	Enters virtual switch submode.
Step 2	`Switch(config-vs-domain)# dual-active detection pagp`	Enables sending of the enhanced PAgP messages.

You must configure trust mode on the port channels that will detect PAgP dual-active detection. By default, trust mode is disabled.

Note If PAgP dual-active detection is enabled, you must place the port channel in administrative down state before changing the trust mode. Use the **shutdown** command in interface configuration mode for the port channel. Remember to use the **no shutdown** command to reactivate the port channels when you are finished configuring trust mode on the port channel.

To configure trust mode on a port channel, perform this task:

	Command	Purpose
Step 1	`Switch(config)# switch virtual domain domain_id`	Enters virtual switch submode.
Step 2	`Switch(config-vs-domain)# dual-active detection pagp trust channel-group group_number`	Enables trust mode for the specified port channel.

This example shows how to enable PAgP dual-active detection:

```
Switch(config)# interface port-channel 20
Switch(config-if)# shutdown
Switch(config-if)# exit
Switch(config)# switch virtual domain 100
Switch(config-vs-domain)# dual-active detection pagp
Switch(config-vs-domain)# dual-active detection pagp trust channel-group 20
Switch(config-vs-domain)# exit
Switch(config)# interface port-channel 20
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

This example shows the error message if you try to enable PAgP dual-active detection when a trusted port channel is not shut down first:

```
Switch(config)# switch virtual domain 100
Switch(config-vs-domain)# dual-active detection pagp
```

This example shows the error message if you try to configure trust mode for a port channel that is not shut down first:

```
Switch(config)# switch virtual domain 100
Switch(config-vs-domain)# dual-active detection pagp trust channel-group 20
Trusted port-channel 20 is not administratively down. To change the pagp dual-active trust
configuration, "shutdown" the port-channel first. Remember to "no shutdown" the
port-channel afterwards.
```

Configuring Fast-Hello Dual-Active Detection

To configure an interface as part of a dual-active detection pair, you need to configure dual-active fast-hello on the interface. Although fast hello dual-active detection is enabled by default, you must configure dual-active interface pairs to act as fast hello dual-active messaging links.

To enable or disable fast-hello dual-active detection, perform this task:

	Command	Purpose
Step 1	`Switch(config)# `**`switch virtual domain`** *`domain_id`*	Enters virtual switch submode.
Step 2	`Switch(config-vs-domain)# `**`dual-active detection fast-hello`**	Enables the fast hello dual-active detection method. **Note** Fast hello dual-active detection is enabled by default.
Step 3	`Switch(config-vs-domain)# `**`exit`**	Exits virtual switch submode.
Step 4	`Switch(config)# `**`interface`** *`type switch/slot/port`*	Selects the interface to configure. **Note** This interface must be directly connected to the other chassis and must not be a VSL link.
Step 5	`Switch(config-if)# `**`dual-active fast-hello`**	Enables fast hello dual-active detection on the interface, automatically removes all other configuration from the interface, and restricts the interface to dual-active configuration commands.
Step 6	`Switch(config-if)# `**`no shutdown`**	Activates the interface.
Step 7	`Switch(config-if)# `**`exit`**	Exits interface configuration mode.
Step 8	`Switch(config)# `**`exit`**	Exits global configuration mode.
Step 9	`Switch)# `**`show run interface`** *`type switch/slot/port`*	Displays status of dual-active fast-hello configuration.

When you configure fast hello dual-active interface pairs, note the following information:

- You can configure a maximum of four interfaces on each chassis to connect with the other chassis in dual-active interface pairs. Attempting to configure more than four interfaces causes an error message to display (and your command is rejected).
- Each interface must be directly connected to the other chassis and must not be a VSL link. We recommend using links from a switching module not used by the VSL.
- Each interface must be a physical port. Logical ports such as an SVI are not supported.
- The fast-hello links are Layer 2 ports.
- Configuring fast hello dual-active mode automatically removes all existing configuration from the interface and restricts the interface to fast hello dual-active configuration commands. It can only be used for “fast-hello” traffic.
- Unidirectional link detection (UDLD) is disabled on fast hello dual-active interface pairs.
- Do not configure fast-hello ports on an oversubscribed line card. Doing so might lead to a loss of fast-hello messages, impacting the functionality of fast-hello based dual-active detection.

This example shows how to configure an interface for fast hello dual-active detection:

```
Switch(config)# switch virtual domain 255
Switch(config-vs-domain)# dual-active detection fast-hello
Switch(config-vs-domain)# exit
Switch(config)# interface fastethernet 1/2/40
Switch(config-if)# dual-active fast-hello
WARNING: Interface FastEthernet1/2/40 placed in restricted config mode. All extraneous
configs removed!
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# exit
Switch# show run interface fastethernet 1/2/40
interface FastEthernet1/2/40
 switchport mode access
 switchport nonegotiate
 dual-active fast-hello
 no switchport
 no ip address
 dual-active fast-hello
end
```

Displaying Dual-Active Detection

To display information about dual-active detection, perform this task:

Command	Purpose
Switch# **show switch virtual** [**dual-active** {**pagp** | **fast-hello** | **summary**} | **link** [**counters** | **detail** | **port-channel** | **ports**] | **redundancy** | **role** | **slot-map**]	Displays information about dual-active detection configuration and status.

This example shows how to display the summary status for dual-active detection:

```
Switch# show switch virtual dual-active summary
Switch(recovery-mode)# show switch virtual dual-act summary
Pagp dual-active detection enabled: Yes
In dual-active recovery mode: Yes
  Triggered by: PagP
  Triggered on Interface: Gi1/3/11
  Received id: e8b7.488e.b7c0
  Expected id: e8b7.488e.b700
```

This example shows how to display the summary status for dual-active detection when recovery is triggered by RRP rather than PagP:

```
Switch# show switch virtual dual-active summary
Switch(recovery-mode)# show switch virtual dual-act summary
Pagp dual-active detection enabled: Yes
In dual-active recovery mode: Yes
  Triggered by: RRP
```

This example shows how to display PAgP status and the channel groups with trust mode enabled:

```
Switch# show pagp dual-active
PAgP dual-active detection enabled: Yes
PAgP dual-active version: 1.1

Channel group 25 dual-active detect capability w/nbrs
Dual-Active trusted group: Yes
          Dual-Active     Partner              Partner   Partner
Port      Detect Capable  Name                 Port      Version
Gi1/3/11  Yes             g9-68                Gi1/11    1.1
Gi2/2/12  Yes             g9-68                Gi1/12    1.1
```

This example shows how to display the status of links configured as fast-hello:

```
Switch# show switch virtual dual-active fast-hello

Executing the command on VSS member switch role = VSS Active, id = 2

Fast-hello dual-active detection enabled: Yes

Fast-hello dual-active interfaces:
Port       Local State                Peer Port
---------------------------------------------------
Gi2/2/11   Dual Active Capable       Gi1/1/5


Executing the command on VSS member switch role = VSS Standby, id = 1

Fast-hello dual-active detection enabled: Yes

Fast-hello dual-active interfaces:
Port       Local State                Peer Port
---------------------------------------------------
Gi1/1/5    Dual Active Capable       Gi2/2/11
```

This example shows how to display the status of packet exchanges between the individual fast-hello links:

```
Switch# show switch virtual dual-active fast-hello counters

Executing the command on VSS member switch role = VSS Active, id = 2

Dual-active fast-hello link counters:
               Tx                 Rx
Port           OK                 OK
--------------------------------------
Gi2/2/11   762       759


Executing the command on VSS member switch role = VSS Standby, id = 1
```

```
Dual-active fast-hello link counters:
               Tx                 Rx
Port           OK                 OK
--------------------------------------
Gi1/1/5    762       759
```

This example shows how to display the status of total packets exchanged between the fast-hello links on the VSS:

```
Switch# show switch virtual dual-active fast-hello packet

Executing the command on VSS member switch role = VSS Active, id = 2

Dual-active fast-hello packet counters:
  SwitchId : 2
  Transmitted:
    total         = 465
  Received:
    total         = 465


Executing the command on VSS member switch role = VSS Standby, id = 1

Dual-active fast-hello packet counters:
  SwitchId : 1
  Transmitted:
    total         = 465
  Received:
    total         = 465
```

In-Service Software Upgrade (ISSU) on a VSS

Topics include

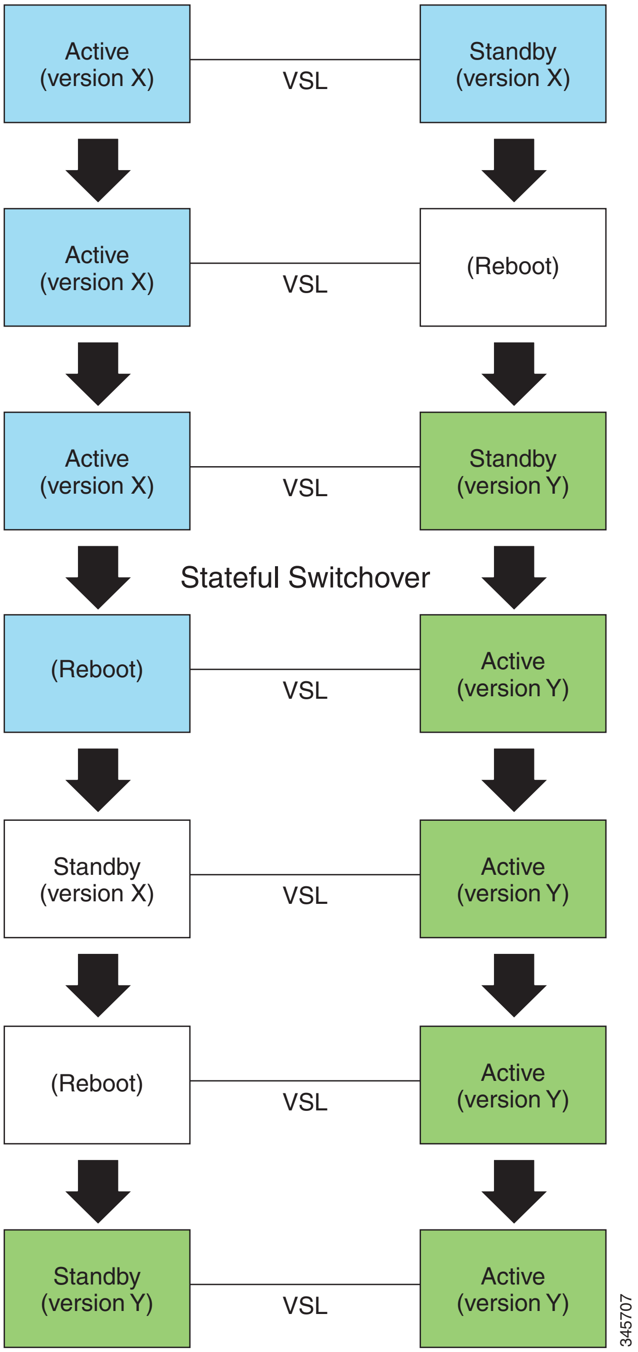
- VSS ISSU Concept, page 1-53
- Traffic and Network Protocol Disruption During ISSU in a VSS, page 1-55
- Related Documents, page 1-55
- Prerequisites to Performing ISSU, page 1-55
- About Performing ISSU, page 1-56
- How to Perform the ISSU Process, page 1-61
- License install and subsequent VSS formation are now complete., page 1-82

VSS ISSU Concept

In a VSS, the supervisor engines on the peer switches maintain an SSO (stateful switchover) relationship between themselves. This facilitates the ability to perform a software upgrade (or downgrade) on both the VSS supervisor engines, one at a time.

Figure 1-9 below depicts (at a conceptual level) the sequence of events that take place when the VSS system is upgraded from software version X to version Y.

Figure 1-9 Upgrading VSS System



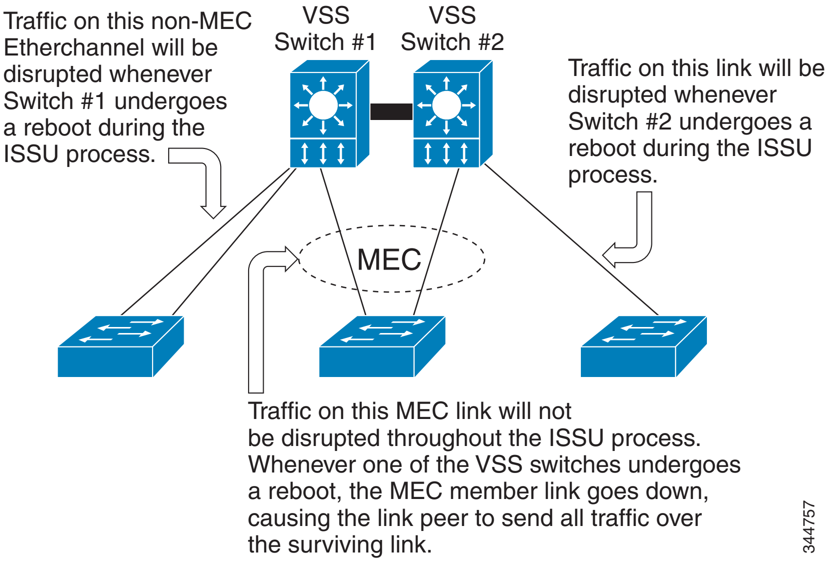
Note that at any given instant, at least one of the switches is Active. The Active switch is in operation, i.e. continues to forward traffic and participate in network control protocols throughout the duration of the upgrade operation.

Traffic and Network Protocol Disruption During ISSU in a VSS

Figure 1-9indicates that both switches in a VSS reboot at some point during the upgrade process.

When a switch reboots, all the network links that terminate on that switch undergo a link-down event. This means that network devices that are connected to the switch that is rebooting will observe a disruption in service, unless the connection is over an MEC that contains at least one link that terminates on the other switch. If a peer device is connected to the VSS over an MEC that has links terminating in both switches, that device will not experience a disruption of service during the software upgrade process. This is illustrated in Figure 1-10.

Figure 1-10 Connecting a Peer Device to VSS to Avoid Service Disruption


Traffic on this non-MEC Etherchannel will be disrupted whenever Switch #1 undergoes a reboot during the ISSU process.
VSS Switch #1
VSS Switch #2
Traffic on this link will be disrupted whenever Switch #2 undergoes a reboot during the ISSU process.
MEC
Traffic on this MEC link will not be disrupted throughout the ISSU process. Whenever one of the VSS switches undergoes a reboot, the MEC member link goes down, causing the link peer to send all traffic over the surviving link.
344757

Related Documents

Related Topic	Document Title
Performing ISSU	*Cisco IOS Software: Guide to Performing In Service Software Upgrades*

Prerequisites to Performing ISSU

Before performing ISSU, you must meet these prerequisites:

- Ensure that the current Cisco IOS XE version running in the system supports ISSU. Also ensure that the target version supports ISSU.

  You can enter various commands on the switch to determine supervisor engine versioning and Cisco IOS XE software compatibility. Alternatively, you can use the ISSU application on Cisco Feature Navigator to determine this.

- The type of the pre- and post-upgrade images must match precisely. Identical. ISSU is not supported from a Universal_lite to a Universal image, or vice versa. ISSU is also not supported from a k9 image to a non-k9 image, or vice versa.
- VSS must be functionally in SSO mode; that is, both switches must be powered up and operational, with one supervisor engine running as the SSO active, and the other as the SSO standby.
- The pre- and post-upgrade Cisco IOS XE software image files must both be available in the local file systems (bootflash, SD card, or USB) of both the Active and the standby supervisor engines before you begin the ISSU process.

  Both supervisor engines should be running the pre-upgrade image, and should have booted from the image location in the local file system. (bootflash, SD card, or USB).

Note The **show version** command can be used to confirm that the supervisor engine has actually booted from the pre-upgrade image location in the local filesystem.

Note Auto-boot must be enabled in the rommon for ISSU to succeed. The config-register value displayed in the output of **show version** can be used to confirm this.

- It is advisable to take measures to mitigate the effects of switch down-time. ISSU in a VSS will result in loss of service on non-MEC links, and peers must be prepared for this. On links connected over MECs, Nonstop Forwarding (NSF) must be configured and working properly. If you do not have NSF enabled, see the *Cisco Nonstop Forwarding* document for further information on how to enable and configure NSF.
- Autoboot is turned on and the current booted image matches the one specified in the BOOT environmental variable. For details on how to configure and verify these, please refer to "Modifying the Boot Field and Using the boot Command" section on page 1-27.
- The **no ip routing** command is not supported - both before starting the ISSU process, and at any time during the ISSU process.
- Save the image on the same partition in both supervisor engines (e.g. if it is saved at slot0: in the active supervisor engine it should be saved at slaveslot0:). Similarly if it is at bootflash: in the active supervisor engine, it should be at savebootflash: in the standby supervisor engine.

About Performing ISSU

Note Do not make any hardware changes while performing ISSU.

Before you perform ISSU, you should understand the following concepts:

- Performing an ISSU Upgrade: Two Methods, page 1-56
- Guidelines for Performing ISSU, page 1-59

Performing an ISSU Upgrade: Two Methods

There are two ways to perform an ISSU upgrade:

- manually using a sequence of four commands

- automatically; using a single command

ISSU using the four-command sequence

The manual ISSU upgrade process involves issuing four distinct ISSU EXEC commands in sequence

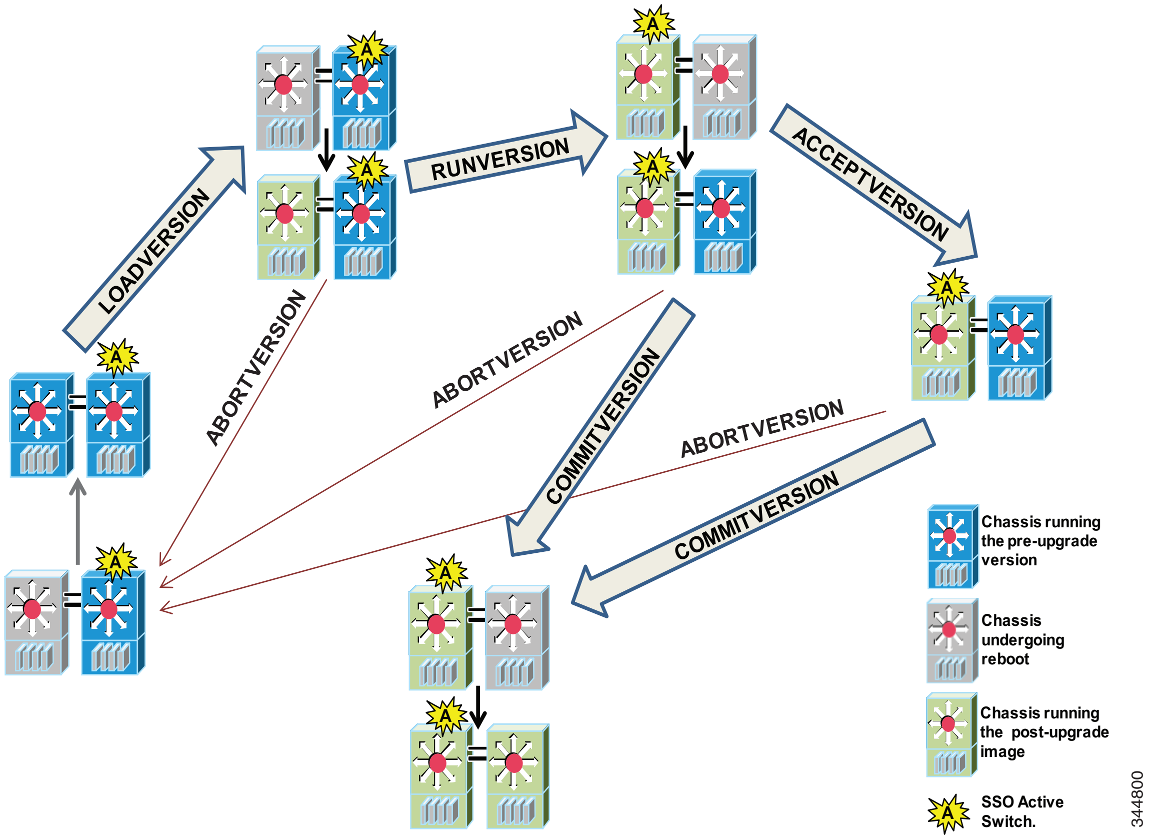
- **issu loadversion**
- **issu runversion**
- **issu acceptversion**
- **issu commitversion**

A fifth command, **issu abortversion**, enables you to abort the ISSU upgrade process at any time, and to revert to the initial system state.

These four commands take the VSS through a series of states that culminate in the Active and standby supervisor engines running the post-upgrade IOS XE image. The VSS continues to operate throughout the entire process; however as explained in Traffic and Network Protocol Disruption During ISSU in a VSS, page 1-55, service is disrupted on network links that terminate on interfaces that reside in the switch that is undergoing a reboot.

Figure 1-11 depicts the states through which the VSS Active and standby supervisor engines progress as the sequence of four commands entered. It also shows the effect of the **issu abortversion** command at any given point during the process.

Figure 1-11 States of VSS Active and Standby during Command Execution



During the ISSU process, several **show** commands are available to evaluate the success of each command before proceeding to the next step.

ISSU using the Single Command Sequence (issu changeversion)

The use of multiple ISSU commands dictates an additional level of care to ensure no service disruption. However, in some scenarios, this upgrade procedure might be cumbersome and of minimal value. A typical example is during a network upgrade that involves performing an ISSU upgrade on a large number of Catalyst 4500 switches. In these cases, we recommend that you first perform the manual (four command) ISSU upgrade procedure on one VSS (possibly in a lab environment) to verify successful upgrade. Then, use the single **issu changeversion** procedure to perform an automatic ISSU on the rest of the Catalyst 4500 switches in the network.

The **issu changeversion** command launches a single-step complete ISSU upgrade cycle. It performs the logic for all four of the standard commands (**issu loadversion**, **issu runversion**, **issu acceptversion**, and **issu commitversion**) without user intervention, streamlining the upgrade through a single CLI step.

Additionally, **issu changeversion** allows the upgrade process to be scheduled for a future time. This enables you to stage a number of systems to perform upgrades sequentially when a potential disruption would be least harmful.

After the standby supervisor engine initializes and the system reaches a terminal state (SSO), the upgrade process is complete and the BOOT variable is permanently written with the new IOS XE software image. Hence, a reset on any RP will keep the system booting the new software image. Console and syslog messages will be generated to notify anyone monitoring the upgrade that the state transition has occurred.

Similar to the normal ISSU upgrade procedure, the in-progress upgrade procedure initiated by the **issu changeversion** command can be aborted with the **issu abortversion** command. If the system detects any problems or detects an unhealthy system during an upgrade, the upgrade might be automatically aborted.

When the **issu runversion** command is entered during the four step manual upgrade process, if any incompatible ISSU clients exist, the upgrade process reports them and their side effects, and allows the user to abort the upgrade. While performing a single-step upgrade process, when the process reaches the runversion state, it will either automatically continue with the upgrade provided the base clients are compatible, or automatically abort because of client incompatibility.

Changeversion: “quick” option

The **issu changeversion** command provides a “quick” option that can reduce the time required to perform the automatic ISSU upgrade. When the **quick** command option is applied, the ISSU upgrade state transition differs from that illustrated in Figure 1-9. With this option, the state progression upto the loadversion stage remains the same as described in the figure, but the runversion and commitversion stages are combined. This progression skips the step in the upgrade procedure that loads the old software version on the new standby (old active) supervisor, thereby reducing the time required for the automatic ISSU upgrade by about a third.

Scheduled Changeversion: “in” and “at” Options

issu changeversion provides **in** and **at** command options that enable you to schedule a future automatic ISSU upgrade.

The **at** command option schedules an automatic ISSU upgrade to begin at a specific time. This option specifies an exact time (*hh*:*mm*, 24 hour format) in the next 24 hours at which the upgrade will occur.

The **in** command option schedules an automatic ISSU upgrade to begin after a certain amount of time has elapsed. This option specifies the number of hours and minutes (*hh*:*mm* format) that must elapse before an upgrade will occur, with a maximum value of 99:59.

Changeversion Deployment Scenario

The typical **issu changeversion** command usage scenario is for experienced users with a large installed base. These users typically validate a new image using a topology and configuration similar to their production network. The validation process should be done using both the existing multi-command process and the new **issu changeversion** command process. Once users certify an IOS XE software image and want to roll it out broadly, they can use the single command process to perform an efficient upgrade of their network.

Aborting an In-Progress Changeversion Procedure

The **issu changeversion** command functionality is designed to perform an ISSU software upgrade without user intervention. However, status messages are displayed to the console as the upgrade transitions through the various states. If any anomalies are noticed during the automatic upgrade, perhaps with peers or other parts of the network, you can use the **issu abortversion** command to manually abort the upgrade at any point in the process prior to the commitversion operation.

Guidelines for Performing ISSU

Be aware of the following guidelines while performing the ISSU process:

- Even with ISSU, it is recommended that upgrades be performed during a maintenance window.
- As explained in Traffic and Network Protocol Disruption During ISSU in a VSS, page 1-55, ISSU on VSS may cause loss of network connectivity to both the VSS switches at some point during the process (although not at the same time). The mitigation steps as explained in that section must be implemented.
- The new features should not be enabled (if they require change of configuration) during the ISSU process.

Note Enabling them will cause the system to enter RPR mode because commands are only supported on the new version.

- In a downgrade scenario, if any feature is not available in the downgrade revision of the Cisco IOS XE software handle, that feature should be disabled prior to initiating the ISSU process.

Compatibility Matrix

ISSU requires additional information to determine compatibility between software versions. Therefore, a compatibility matrix is defined that contains information about other IOS XE software image with respect to the one in question.

This compatibility matrix represents the compatibility of two software versions, one running on the active and the other on the standby supervisor engine, and to allow the system to determine the highest operating mode it can achieve. Incompatible versions will not be able to progress to SSO operational mode. Because SSO is a pre-requisite for a VSS, incompatibility will also lead to the loss of VSS relationship between the supervisors engines in the two switches.

The compatibility matrix represents the compatibility relationship a Cisco IOS XE software image has with all of the other Cisco IOS XE software versions within the designated support window (for example, all of those software versions the IOS XE software image "knows" about) and is populated and released with every IOS XE software image. The matrix stores compatibility information between its own release and prior releases. It is always the newest release that contains the latest information about compatibility with existing releases in the field. The compatibility matrix is available within the Cisco IOS XE software image and on Cisco.com so that users can determine in advance whether a successful upgrade can be achieved using the ISSU process.

You can perform the ISSU process when the old and new Cisco IOS XE software are compatible. The compatibility matrix information stores the compatibility among releases as follows:

- Compatible—The base-level system infrastructure and all optional HA-aware subsystems are compatible. An in-service upgrade or downgrade between these versions will succeed with minimal service impact. The matrix entry designates the images to be compatible (C).
- Base-level compatible—One or more of the optional HA-aware subsystems is not compatible. An in-service upgrade or downgrade between these versions will succeed; however, some subsystems will not be able to maintain state always during the transition from the old to the new version of Cisco IOS XE. The matrix entry designates the images to be base-level compatible (B).
- Incompatible—A core set of system infrastructure exists in Cisco IOS XE that must be able to interoperate in a stateful manner for SSO to function correctly. If any of these required features or subsystems is not interoperable, then the two versions of the Cisco IOS XE software image are declared to be incompatible. An in-service upgrade or downgrade between these versions is not possible. The matrix entry designates the images to be incompatible (I). The system operates in RPR mode during the upgrade process when the versions of Cisco IOS XE at the active and standby supervisor engines are incompatible.
- Cisco IOS XE determines the compatibility between the active and the standby IOS XE software dynamically during Standby boot up. The matrix is represented by "x".

To display the compatibility matrix data between two software versions on a given system, enter the **show issu comp-matrix stored** command.

Note This command is useful *only for verification purpose*s because it is available *only after* the ISSU process has started. You might want to check the compatibility matrix prior to starting ISSU. Use the Feature Navigator to obtain the needed information:

http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp

Compatibility Verification Using Cisco Feature Navigator

The ISSU application on Cisco Feature Navigator allows you to:

- Select a specific software bundle.
- Identify which software images are compatible with the selected software image.
- Compare two IOS XE software images and understand the compatibility level of the software images (that is, compatible, base-level compatible, and incompatible), or dynamically determined.
- Compare two software images and see the client compatibility for each ISSU client.
- Provide links to release notes for the software image.

How to Perform the ISSU Process

Unlike SSO, which is a mode of operation for the device and a prerequisite for performing ISSU, the ISSU process is a series of steps performed while the switch is in operation. The steps result in an upgrade to new or modified Cisco IOS XE software, and have a minimal impact to traffic.

Note For an illustration of the process flow for ISSU, refer to Figure 1-11.

This section includes the following topics:

- Verifying the ISSU Software Installation, page 1-61
- Verifying Redundancy Mode Before Beginning the ISSU Process, page 1-62
- Verifying the ISSU State Before Beginning the ISSU Process, page 1-63
- ISSU using the Four-command Sequence: Step 1 (loadversion), page 1-65
- ISSU using the Four-command Sequence: Step 2 (runversion), page 1-66
- ISSU using the Four Command Sequence: Step 3 (acceptversion), page 1-68
- ISSU using the Four Command Sequence: Step 4 (commitversion), page 1-69
- Using changeversion to Automate an ISSU Upgrade, page 1-70
- Aborting a Software Upgrade During ISSU, page 1-76
- Configuring the Rollback Timer to Safeguard Against Upgrade Issues, page 1-77
- The ISSU Compatibility Matrix, page 1-79

Verifying the ISSU Software Installation

During the ISSU process, there are five valid states: disabled, init, load version, run version, and system reset. Use the **show issu state** command to obtain the current ISSU state:

- Disabled state—The state for the standby supervisor engine while this supervisor engine is resetting.
- Init state—The initial state for two supervisor engines, one active and one standby, before the ISSU process is started. It is also the final state after the ISSU process completes.
- Load version (LV) state—The standby supervisor engine is loaded with the new version of Cisco IOS XE software.
- Run version (RV) state—The **issu runversion** command forces the switchover of the supervisor engines. The newly active supervisor engine runs the new Cisco IOS XE software image.
- While running ISSU, if both supervisor engines are reset (because of a power outage, for example), the ISSU context is lost and the system returns to the Init state. Both supervisor engines return to the old software.

You can verify the ISSU software upgrade by entering **show** commands to provide information on the state of the during the ISSU process:

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch# show issu state [detail]`	Displays the current state of the ISSU process.

	Command or Action	Purpose
Step 3	`Switch# show redundancy`	Displays current or historical status, mode, and related redundancy information about the device.
Step 4	`Switch# show switch virtual`	Identifies which switch of the VSS is currently performing the Active role, and which switch the Standby.

This example shows how to display the state and the current status of the supervisor engine during the ISSU process:

```
Switch> enable
Switch# show issu state
Switch# show redundancy
Switch# show switch virtual
```

Verifying Redundancy Mode Before Beginning the ISSU Process

Before you begin the ISSU process, verify that the VSS is operating correctly; one supervisor engine operates as the SSO Active and the peer supervisor engine in the other switch operating as the SSO Hot Standby.

The following example displays verification that the system operating correctly as a VSS. Slot 1/1 (the supervisor engine in slot 1 of Switch 1) is the active supervisor engine, and Slot 2/1 (the supervisor engine in slot 1 of Switch 2) is the standby supervisor engine.

```
Switch# show redundancy states
         my state = 13 -ACTIVE
       peer state =  8 -STANDBY HOT
             Mode = Duplex
             Unit = Primary
          Unit ID = 1

  Redundancy Mode (Operational) = Stateful Switchover
   Redundancy Mode (Configured) = Stateful Switchover
               Redundancy State = Stateful Switchover
                      Manual Swact = enabled

   Communications = Up

     client count = 77
   client_notification_TMR = 240000 milliseconds
            keep_alive TMR = 9000 milliseconds
          keep_alive count = 0
      keep_alive threshold = 18
             RF debug mask = 0

Switch# show redundancy
Redundant System Information :

------------------------------
       Available system uptime = 11 minutes
Switchovers system experienced = 0
              Standby failures = 0
        Last switchover reason = none

                 Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
     Operating Redundancy Mode = Stateful Switchover
              Maintenance Mode = Disabled
                Communications = Up
```

```
Current Processor Information :
------------------------------
               Active Location = slot 1/1
        Current Software state = ACTIVE
       Uptime in current state = 9 minutes
                 Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.33 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 09-Aug
               BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;
        Configuration register = 0x2102

Peer Processor Information :
------------------------------
              Standby Location = slot 2/1
        Current Software state = STANDBY HOT
       Uptime in current state = 8 minutes
                 Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.33 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 09-Au
               BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;
        Configuration register = 0x2102



Switch# show switch virtual
Switch mode                  : Virtual Switch
Virtual switch domain number : 16
Local switch number          : 1
Local switch operational role: Virtual Switch Active
Peer switch number           : 2
Peer switch operational role : Virtual Switch Standby
Switch#
```

Verifying the ISSU State Before Beginning the ISSU Process

Ensure that both the supervisor engines are configured to auto-boot, and that they have currently been booted from the pre-upgrade image residing on the local file system. This is verified by the values of the BOOT variable and the configuration register (refer to the sample output of **show redundancy** command in the previous section).

Ensure that the active and standby supervisor engines are up and in ISSU Init state and that both supervisor engines are running the same current image.

The following example displays the ISSU state before the process begins:

```
Switch# show issu state detail
                               Slot = 1
                           RP State = Active
                         ISSU State = Init
                     Operating Mode = Stateful Switchover
                      Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin
          Pre-ISSU (Original) Image = N/A
         Post-ISSU (Targeted) Image = N/A
```

```
                               Slot = 11
                           RP State = Standby
                         ISSU State = Init
                     Operating Mode = Stateful Switchover
                      Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin
          Pre-ISSU (Original) Image = N/A
         Post-ISSU (Targeted) Image = N/A
```

Note that the Standby slot number is reported as 11, which is a Virtual Slot number that corresponds to physical slot 1 on Switch 2. The correspondence between the Virtual Slot number and the physical location of the slot can be determined using the **show switch virtual slot-map** command, as shown in the following example:

```
Switch# show switch virtual slot-map
Virtual Slot to Remote Switch/Physical Slot Mapping Table:

Virtual    Remote      Physical   Module
Slot No    Switch No   Slot No    Uptime
---------+-----------+----------+----------
  1         1             1        00:33:04
  2         1             2        00:32:50
  3         1             3        00:32:36
  4         1             4          -
  5         1             5          -
  6         1             6          -
  7         1             7          -
  8         1             8          -
  9         1             9          -
  10        1             10         -
  11        2             1        00:31:14
  12        2             2        00:33:33
  13        2             3        00:33:33
  14        2             4          -
  15        2             5          -
  16        2             6          -
  17        2             7          -
  18        2             8          -
  19        2             9          -
  20        2             10         -
Switch#
```

The new version of the Cisco IOS XE software must be present on both of the supervisor engines. The directory information displayed for each of the supervisor engines shows that the new version is present.

```
Switch# dir bootflash:
Directory of bootflash:/

29122  -rw-   119519232  Aug 13 2012 19:13:14 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
29125  -rw-   119286584  Aug 13 2012 22:30:02 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin

820875264 bytes total (581672960 bytes free)
Switch# dir slavebootflash:
Directory of slavebootflash:/
```

```
58370  -rw-   119286584  Aug 14 2012 11:25:38 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin
58372  -rw-   119519232  Aug 14 2012 11:40:47 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin

822910976 bytes total (583716864 bytes free)
Switch#
```

ISSU using the Four-command Sequence: Step 1 (loadversion)

This task describes the first step of the ISSU four-command sequence, loadversion, wherein the standby supervisor engine is loaded with the post-upgrade image.

Please ensure that you have read the Prerequisites to Performing ISSU, page 1-55 section, and implemented the appropriate steps.

Perform the following steps at the active supervisor engine:

	Command or Action	Purpose
Step 1	`Switch>` **`enable`**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch#` **`issu loadversion`** [*`active-slot`*] *`active-image-new`* [*`standby-slot`*] *`standby-image-new`*	Starts the ISSU process and (optionally) overrides the automatic rollback when the new Cisco IOS XE software version is detected to be incompatible. It may take several minutes after entering the **issu loadversion** command for Cisco IOS XE software to load onto the standby supervisor engine and for the standby supervisor engine to transition to SSO mode. This causes the standby supervisor engine to reload with the new software image. If used, the *active-slot* and *standby-slot* numbers should be specified as Virtual Slot numbers. Use the **show switch virtual slot-map** command to determine the correspondence between Virtual slot numbers and their physical locations.
Step 3	`Switch#` **`show issu state`** [**`detail`**]	Displays the state of ISSU during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded and is in SSO mode. It may take several minutes after entering the **issu loadversion** command for Cisco IOS XE software to load onto the standby supervisor engine and the standby supervisor engine to transition to SSO mode. If you enter the **show issu state** command too quickly, you may not see the information you need.
Step 4	`Switch#` **`show redundancy`** [**`states`**]	Displays redundancy facility state information.

This example shows how to start the ISSU process, boot the standby supervisor engine in the Standby Hot state, and load the standby supervisor engine slot (6) with the new IOS XE software image:

```
Switch> enable
Switch# issu loadversion 1 bootflash:new_image 11 slavebootflash:new_image
```

```
%issu loadversion executed successfully, Standby is being reloaded
Switch# show issu state detail
                               Slot = 1
                           RP State = Active
                         ISSU State = Load Version
                     Operating Mode = Stateful Switchover
                      Current Image = bootflash:old_image
          Pre-ISSU (Original) Image = bootflash:old_image
         Post-ISSU (Targeted) Image = bootflash:new_image

                               Slot = 11
                           RP State = Standby
                         ISSU State = Load Version
                     Operating Mode = Stateful Switchover
                      Current Image = bootflash:new_image
          Pre-ISSU (Original) Image = bootflash:old_image
         Post-ISSU (Targeted) Image = bootflash:new_image

Switch# show redundancy states
         my state = 13 -ACTIVE
       peer state =  8 -STANDBY HOT
             Mode = Duplex
             Unit = Primary
          Unit ID = 1

  Redundancy Mode (Operational) = Stateful Switchover
   Redundancy Mode (Configured) = Stateful Switchover
               Redundancy State = Stateful Switchover
                   Manual Swact = enabled

   Communications = Up

     client count = 474
   client_notification_TMR = 240000 milliseconds
            keep_alive TMR = 9000 milliseconds
          keep_alive count = 1
      keep_alive threshold = 18
             RF debug mask = 0
```

ISSU using the Four-command Sequence: Step 2 (runversion)

This task describes the second step of the ISSU four-command sequence, runversion, wherein a switchover occurs and the standby supervisor engine, which is now loaded with the post-upgrade image, takes over as the new Active.

At the end of the loadversion step, the following message is logged:

```
*Aug 14 13:07:08.240: %INSTALLER-7-ISSU_OP_SUCC: Peer state is [STANDBY SSO]; Please issue
the runversion command
```

Now, you are ready to proceed to the next step.

Perform the following steps at the active supervisor engine.

	Command or Action	Purpose
Step 1	`Switch>` **`enable`**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch#` **`issu runversion`** `[`*`standby-slot`*`] [`*`standby-image-new`*`]]`	Forces a switchover from the active to the standby supervisor engine and reloads the former active (current standby) supervisor engines with the old IOS XE image. As with any SSO switchover, you are prompted to save the running configuration if you have changed it. Respond as appropriate. When you enter the **issu runversion** command, an SSO switchover will be performed, and NSF procedures will be invoked if so configured.
Step 3	`Switch#` **`show issu state`** `[`**`detail`**`]`	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that a switchover occurs to slot 11.
Step 4	`Switch#` **`show redundancy`** `[`**`states`**`]`	Displays redundancy facility state information.

This example shows how to cause a switchover to the former standby supervisor engine (slot 11), reset the former active supervisor engine and reload it with the old IOS XE software image so it becomes the standby supervisor engine:

```
Switch> enable
Switch# issu runversion 11 slavebootflash:new_image
%issu runversion initiated successfully
```

A switchover happens at this point. At the new active supervisor engine, do the following after old active supervisor engine comes up as standby.

```
Switch# show issu state detail
                               Slot = 11
                           RP State = Active
                         ISSU State = Run Version
                     Operating Mode = Stateful Switchover
                      Current Image = bootflash:new_image
         Pre-ISSU (Original) Image = bootflash:old_image
        Post-ISSU (Targeted) Image = bootflash:new_image

                               Slot = 1
                           RP State = Standby
                         ISSU State = Run Version
                     Operating Mode = Stateful Switchover
                      Current Image = bootflash:old_image
         Pre-ISSU (Original) Image = bootflash:old_image
        Post-ISSU (Targeted) Image = bootflash:new_image
```

Note The new active supervisor engine is now running the new version of software, and the standby supervisor engine is running the old version of software and is in the standby hot state.

```
Switch# show redundancy states
         my state = 13 -ACTIVE
       peer state =  8 -STANDBY HOT
             Mode = Duplex
             Unit = Primary
          Unit ID = 11

  Redundancy Mode (Operational) = Stateful Switchover
   Redundancy Mode (Configured) = Stateful Switchover
               Redundancy State = Stateful Switchover
                   Manual Swact = enabled

   Communications = Up

     client count = 474
   client_notification_TMR = 240000 milliseconds
            keep_alive TMR = 9000 milliseconds
          keep_alive count = 0
      keep_alive threshold = 18
             RF debug mask = 0
```

Once Runversion has completed, the new active supervisor engine will be running the new version of software and the previously active supervisor engine will now become the standby supervisor engine. The standby supervisor engine will be reset and reloaded, but it will remain on the previous version of software and come back online in Standby hot status.

Use the **show redundancy**, **show redundancy states**, and **show issu state** [**detailed**] commands described previously to verify that the standby supervisor engine is running the pre-upgrade version and that the Active is running the post-upgrade version.

ISSU using the Four Command Sequence: Step 3 (acceptversion)

This step is optional. It is needed only if you wish to stop the ISSU rollback timer. Otherwise you may proceed to the next step (**commitversion**)

Cisco IOS XE software maintains an ISSU rollback timer to safeguard against an upgrade that may leave the new active supervisor engine in a state in which communication with the standby supervisor engine is severed. By default, this duration is 45 minutes. If the **commitversion** command is not applied before the rollback timer duration expires, the VSS reverts to the pre-upgrade version.

The **acceptversion** command stops the rollback timer. This means that you can maintain the system in the current state (runversion, with the post-upgrade version running on the active supervisor engine, and pre-upgrade image running on the standby supervisor engine) for an extended duration, and proceed to the commitversion state only when you are satisfied with the behavior of the post-upgrade software version.

This optional task describes how to stop the rollback timer.

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch# issu acceptversion [active-slot] [active-image-new]]`	Halts the rollback timer and ensures the new Cisco IOS XE ISSU process is not automatically aborted during the ISSU process. Enter the **issu acceptversion** command within the time period specified by the rollback timer to acknowledge that the supervisor engine has achieved connectivity to the outside world; otherwise, the ISSU process is terminated, when the rollback timer expires, and the system reverts to the previous version of Cisco IOS XE software by switching to the standby supervisor engine.
Step 3	`Switch# show issu rollback-timer`	Displays the amount of time left before an automatic rollback will occur.

This example displays the Timer before you stop it. In the following example, the “Automatic Rollback Time” information indicates the amount of time remaining before an automatic rollback will occur.

```
Switch> enable
Switch# show issu rollback-timer
     Rollback Process State = 00:31:09 remaining
   Configured Rollback Time = 00:45:00

Switch# issu acceptversion 611 bootflash:new_image
% Rollback timer stopped. Please issue the commitversion command.
Switch# show issu rollback-timer
     Rollback Process State = Not in progress
   Configured Rollback Time = 00:45:00
```

ISSU using the Four Command Sequence: Step 4 (commitversion)

The commitversion step reloads the standby supervisor engine with the post-upgrade image.

Perform the following steps at the active supervisor engine:

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch# issu commitversion [standby-slot] [standby-image-new]`	Allows the new Cisco IOS XE software image to be loaded into the standby supervisor engine.
Step 3	`Switch# show redundancy [states]`	Displays redundancy facility state information.
Step 4	`Switch# show issu state [detail]`	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded with the new image.

This example shows how to reset and reload the current standby supervisor engine (slot 1) with the new Cisco IOS XE software version. After you enter the **commitversion** command, the standby supervisor engine boots in the Standby Hot state.

```
Switch> enable
Switch# issu commitversion 1 slavebootflash:new_image
%issu commitversion executed successfully
```

As in prior states, the **show redundancy**, **show redundancy states**, **show issu state** [**detailed**], and **show switch virtual** commands can be used to verify that the VSS has reached the desired state.

At the end of the commitversion state, the ISSU process has completed. At this stage, any further Cisco IOS XE software version upgrade or downgrade will require that a new ISSU process be invoked anew.

Using changeversion to Automate an ISSU Upgrade

This task describes how to use the **issu changeversion** command to perform a one step ISSU upgrade.

Please ensure that you have read Prerequisites to Performing ISSU, page 1-55, and implemented the appropriate steps.

Perform the following steps at the active supervisor engine:

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch# issu changeversion [active-slot active-image-new]] [standby-slot [standby-image-new]] [at hh:mm | in hh:mm] [quick]`	Initiates a single-step complete upgrade process cycle. Performs the logic of the four standard commands (issu loadversion, issu runversion, issu acceptversion, and issu commitversion) without user intervention. *active-slot*—Defines the active slot number (the Virtual slot number). Use the **show switch virtual slot-map** command to determine the virtual slot number from the physical slot number. *new-image*—Specifies IOS XE image URL to be upgraded to. *standby-slot*—Defines the standby slot number (the Virtual slot number). *standby-image—Specifies the* standby IOS XE image URL. **at** *hh:mm*—Schedules an ISSU upgrade to begin in the future. Provides an exact time (*hh*:*mm*, 24 hour format) in the next 24 hours when the upgrade will occur. **in** *hh:mm*—Schedules an ISSU upgrade to begin in the future. Provides the number of hours and minutes (*hh*:*mm* format) that will elapse before an upgrade will occur (99:59 max). **quick**—Upon switchover, boots the standby supervisor engine with the new, rather than old, image for faster upgrade.
Step 3	`Switch# show issu state [detail]`	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded and is in SSO mode.
Step 4	`Switch# show redundancy [states]`	Displays redundancy facility state information.

This example shows how to initiate an ISSU upgrade process using the **issu changeversion** command. The outputs of the **show switch virtual**, **show issu state detai**l, **show redundancy**, and **show redundancy states** commands are included to show the supervisor state before and after the upgrade procedure.

```
Switch> enable
Switch# show switch virtual
Switch mode                  : Virtual Switch
Virtual switch domain number : 16
Local switch number          : 1
Local switch operational role: Virtual Switch Active
Peer switch number           : 2
Peer switch operational role : Virtual Switch Standby
Switch#
Switch#show redundancy states
         my state = 13 -ACTIVE
       peer state =  8 -STANDBY HOT
             Mode = Duplex
             Unit = Primary
          Unit ID = 1

  Redundancy Mode (Operational) = Stateful Switchover
   Redundancy Mode (Configured) = Stateful Switchover
               Redundancy State = Stateful Switchover
                      Manual Swact = enabled

   Communications = Up

     client count = 74
   client_notification_TMR = 240000 milliseconds
            keep_alive TMR = 9000 milliseconds
          keep_alive count = 0
      keep_alive threshold = 18
             RF debug mask = 0

Switch# show redundancy
Redundant System Information :

------------------------------
       Available system uptime = 3 hours, 50 minutes
Switchovers system experienced = 2
              Standby failures = 1
        Last switchover reason = active unit removed

                 Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
     Operating Redundancy Mode = Stateful Switchover
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
------------------------------
               Active Location = slot 1/1
        Current Software state = ACTIVE
       Uptime in current state = 45 minutes
                 Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.33 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 09-Aug
               BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;bootflash:cat4500e-
universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin,12;
```

```
        Configuration register = 0x2102

Peer Processor Information :
------------------------------
              Standby Location = slot 2/1
        Current Software state = STANDBY HOT
       Uptime in current state = 25 minutes
                 Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.33 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 09-Au
               BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;bootflash:cat4500e-
universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin,12;
        Configuration register = 0x2102



Switch# show issu state detail
                               Slot = 1
                           RP State = Active
                         ISSU State = Init
                     Operating Mode = Stateful Switchover
                      Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin
          Pre-ISSU (Original) Image = N/A
         Post-ISSU (Targeted) Image = N/A

                               Slot = 11
                           RP State = Standby
                         ISSU State = Init
                     Operating Mode = Stateful Switchover
                      Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin
          Pre-ISSU (Original) Image = N/A
         Post-ISSU (Targeted) Image = N/A


Switch# show switch virtual slot-map
Virtual Slot to Remote Switch/Physical Slot Mapping Table:

Virtual    Remote      Physical   Module
Slot No    Switch No   Slot No    Uptime
---------+-----------+----------+----------
  1         1              1        00:44:19
  2         1              2        00:44:05
  3         1              3        00:43:49
  4         1              4          -
  5         1              5          -
  6         1              6          -
  7         1              7          -
  8         1              8          -
  9         1              9          -
  10        1              10         -
  11        2              1        00:26:40
  12        2              2        00:44:48
  13        2              3        00:44:48
  14        2              4          -
  15        2              5          -
  16        2              6          -
  17        2              7          -
  18        2              8          -
  19        2              9          -
  20        2              10         -
Switch# dir bootflash:
Directory of bootflash:/

29122  -rw-   119519232  Aug 13 2012 19:13:14 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
29125  -rw-   119286584  Aug 13 2012 22:30:02 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin

820875264 bytes total (581648384 bytes free)
Switch# dir slavebootflash:
Directory of slavebootflash:/

58372  -rw-   119519232  Aug 14 2012 11:40:47 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
58370  -rw-   119286584  Aug 14 2012 11:25:38 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin

822910976 bytes total (583688192 bytes free)
Switch# issu changeversion
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
%% 'issu changeversion' is now executing 'issu loadversion'
% issu loadversion executed successfully, Standby is being reloaded


%% changeversion finished executing loadversion, waiting for standby to reload and reach
SSO ...
```

Switch 2 goes down, reboots with the post-upgrade image, then reaches SSO Hot Standby state.

```
...

*Aug 14 15:45:45.931: %RF-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)
*Aug 14 15:48:45.958: %INSTALLER-7-ISSU_OP_SUCC:  issu changeversion is now executing
'issu runversion'Please stand by while rebooting the system...
```

A Stateful Switchover occurs. Switch 2 takes over as the Active switch. Switch 1 goes down, then reboots (still with the pre-upgrade image) and reaches SSO Hot Standby state.

(From this point on, the console logs are gathered on Switch 2)

```
*Aug 14 15:54:49.164: %INSTALLER-7-ISSU_OP_SUCC:  issu changeversion is now executing
'issu commitversion'
```

Switch 1 goes down again, then boots up (this time with the post-upgrade image), and comes up as SSO Hot Standby

```
Switch# show switch virtual
Switch mode                  : Virtual Switch
Virtual switch domain number : 16
Local switch number          : 2
Local switch operational role: Virtual Switch Active
Peer switch number           : 1
Peer switch operational role : Virtual Switch Standby

Switch# show switch virtual slot-map
Virtual Slot to Remote Switch/Physical Slot Mapping Table:

Virtual    Remote      Physical   Module
Slot No    Switch No   Slot No    Uptime
---------+-----------+----------+----------
  1         1              1        00:01:21
  2         1              2        00:19:12
  3         1              3        00:19:12
  4         1              4          -
  5         1              5          -
  6         1              6          -
  7         1              7          -
  8         1              8          -
  9         1              9          -
  10        1              10         -
  11        2              1        00:18:43
  12        2              2        00:18:17
  13        2              3        00:18:16
  14        2              4          -
  15        2              5          -
  16        2              6          -
  17        2              7          -
  18        2              8          -
  19        2              9          -
  20        2              10         -
Switch#show issu state detail
                               Slot = 11
                           RP State = Active
                         ISSU State = Init
                     Operating Mode = Stateful Switchover
                      Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
          Pre-ISSU (Original) Image = N/A
         Post-ISSU (Targeted) Image = N/A

                               Slot = 1
                           RP State = Standby
                         ISSU State = Init
                     Operating Mode = Stateful Switchover
                      Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
          Pre-ISSU (Original) Image = N/A
         Post-ISSU (Targeted) Image = N/A


Switch# show redundancy states
         my state = 13 -ACTIVE
       peer state =  8 -STANDBY HOT
             Mode = Duplex
             Unit = Primary
          Unit ID = 11

  Redundancy Mode (Operational) = Stateful Switchover
   Redundancy Mode (Configured) = Stateful Switchover
               Redundancy State = Stateful Switchover
                      Manual Swact = enabled

   Communications = Up

     client count = 74
   client_notification_TMR = 240000 milliseconds
            keep_alive TMR = 9000 milliseconds
          keep_alive count = 0
      keep_alive threshold = 18
             RF debug mask = 0

Switch# show redundancy
Redundant System Information :

------------------------------
       Available system uptime = 4 hours, 16 minutes
Switchovers system experienced = 3
```

```
              Standby failures = 1
        Last switchover reason = active unit removed

                 Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
     Operating Redundancy Mode = Stateful Switchover
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
------------------------------
               Active Location = slot 2/1
        Current Software state = ACTIVE
       Uptime in current state = 21 minutes
                 Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.34 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Fri 10-Aug
               BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin,12;bootflash:cat4500e-
universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;
        Configuration register = 0x2102

Peer Processor Information :
------------------------------
              Standby Location = slot 1/1
        Current Software state = STANDBY HOT
       Uptime in current state = 1 minute
                 Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.34 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Fri 10-Au
               BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin,12;bootflash:cat4500e-
universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;
        Configuration register = 0x2102
```

The following example shows how to use issu changeversion with the "at" command option to schedule an ISSU upgrade procedure to automatically start at the specified time. This example specifies that the ISSU upgrade should be started at 16:30 (24 hour format).

```
Switch> enable
Switch# issu changeversion 1 bootflash:y.bin 11 slavebootflash:y at 16:30
% 'issu changeversion' was executed at [ Aug 12 16:27:43 ].
% The planned ISSU changeversion is to occur in (hh:mm:ss) [ 00:03:00 ] at [ Apr 12
16:30:43 ].
% Current system time: [ Aug 12 16:27:43 ]
% Planned upgrade image: bootflash:y.bin
% To cancel the planned upgrade, please execute 'issu abortversion'

Switch# show issu state detail
                               Slot = 1
                           RP State = Active
                         ISSU State = Init
                        Changeversion = TRUE
                     Operating Mode = Stateful Switchover
                      Current Image = bootflash:x.bin
          Pre-ISSU (Original) Image = N/A
         Post-ISSU (Targeted) Image = N/A

                               Slot = 11
                           RP State = Standby
```

```
                          ISSU State = Init
                       Changeversion = TRUE
                      Operating Mode = Stateful Switchover
                       Current Image = bootflash:x.bin
         Pre-ISSU (Original) Image = N/A
        Post-ISSU (Targeted) Image = N/A
```

Aborting a Software Upgrade During ISSU

You can abort the ISSU process at any stage manually (prior to entering the **issu commitversion** command) by entering the **issu abortversion** command. The **issu abortversion** command may also be issued after entering the **issu changeversion** command while the automatic ISSU upgrade is still in progress. The ISSU process also aborts on its own if the software detects a failure.

Note If you enter the **issu abortversion** command before the standby supervisor engine becomes hot, the traffic might be disrupted.

If you abort the process after you issue the **issu loadversion** command, the standby supervisor engine is reset and reloaded with the original software.

If the process is aborted after you enter either the **issu runversion** or **issu acceptversion** command, then a second switchover is performed to the new standby supervisor engine that is still running the original software version. The supervisor engine that had been running the new software is reset and reloaded with the original software version.

Note Ensure that the standby supervisor is fully booted *before* issuing the **abortversion** command on an active supervisor engine.

The following task describes how to abort the ISSU process before you complete the ISSU process with the **issu commitversion** command.

Perform the following task on the active supervisor engine:

	Command or Action	Purpose
Step 1	`Switch> `**`enable`**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch# `**`issu abortversion`**` [`*`active slot`*` [`*`active-image-new`*`]]`	Cancels the ISSU upgrade or downgrade process in progress and restores the switch to its state before the process had started.

This example shows how to abort the ISSU process on slot number 11, the slot for the current active supervisor engine. In this example, the ISSU upgrade process is in the Runversion state when the issu abortversion command is entered:

```
Switch> enable
Switch# show issu state detail
                                Slot = 11
                            RP State = Active
                          ISSU State = Run Version
                      Operating Mode = Stateful Switchover
                       Current Image = bootflash:x.bin
         Pre-ISSU (Original) Image = bootflash:y.bin
         Post-ISSU (Targeted) Image = bootflash:x.bin

                              Slot = 1
                          RP State = Standby
                        ISSU State = Run Version
                    Operating Mode = Stateful Switchover
                     Current Image = bootflash:y.bin
          Pre-ISSU (Original) Image = bootflash:y.bin
         Post-ISSU (Targeted) Image = bootflash:x.bin

Switch# issu abortversion 11
% issu abortversion initiated successfully
Switch# show issu state detail

                              Slot = 1
                          RP State = Active
                        ISSU State = Init
                    Operating Mode = Stateful Switchover
                     Current Image = bootflash:y.bin
          Pre-ISSU (Original) Image = N/A
         Post-ISSU (Targeted) Image = N/A

                              Slot = 11
                          RP State = Standby
                        ISSU State = Init
                    Operating Mode = Stateful Switchover
                     Current Image = bootflash:y.bin
          Pre-ISSU (Original) Image = N/A
         Post-ISSU (Targeted) Image = N/A

Switch#
```

Configuring the Rollback Timer to Safeguard Against Upgrade Issues

Cisco IOS XE software maintains an ISSU rollback timer, to safeguard against an upgrade that may leave the new active supervisor engine in a state in which communication with the standby supervisor engine is severed.

You may want to configure the rollback timer to fewer than 45 minutes (the default) so that you need not wait in case the new software is not committed or the connection to the switch was lost while it was in runversion mode. Conversely, you may want to configure the rollback timer to more than 45 minutes in order to have enough time to verify the operation of the new Cisco IOS XE software before committing the new software image.

The ISSU rollback timer kicks in immediately after **issu run version** is entered so that the minimum value configured should be more than the time required for a chassis reload. Else, the process fails.

Note The valid timer value range is from 0 to 7200 seconds (two hours). A value of 0 seconds disables the rollback timer.

Once you are satisfied that the new image at the active supervisor engine has been successful and you want to remain in the current state, you may indicate acceptance by issuing the **issu acceptversion** command, which stops the rollback timer.

Issuing the **issu commitversion** command at this stage is equal to entering both the **issu acceptversion** and the **issu commitversion** commands. Use the **issu commitversion** command if you do not intend to run in the current state for a period of time and are satisfied with the new software version.

Note The rollback timer can be configured only in the ISSU Init state.

This task explains how to configure the rollback timer:

	Command or Action	Purpose
Step 1	`Switch>` **`enable`**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Switch(config)#` **`issu set rollback-timer`** *`value`*	Configures the rollback timer value, which can range from 0 to 7200.
Step 4	`Switch(config)#` **`exit`**	Returns the user to privileged EXEC mode.
Step 5	`Switch#` **`show issu rollback-timer`**	Displays the current setting of the ISSU rollback timer.

This example shows how to set the rollback timer to 3600 seconds:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# issu set rollback-timer 3600
% Rollback timer value set to [ 3600 ] seconds

Switch(config)# exit

Switch# show issu rollback-timer
        Rollback Process State = Not in progress
      Configured Rollback Time = 60:00
```

The Rollback Timer cannot be set in loadversion or runversion state, as the following example illustrates:

```
Switch# show issu state detail
                               Slot = 1
                           RP State = Active
                         ISSU State = Load Version
                     Operating Mode = Stateful Switchover
                      Current Image = bootflash:old_image
          Pre-ISSU (Original) Image = bootflash:old_image
         Post-ISSU (Targeted) Image = bootflash:new_image

                               Slot = 11
                           RP State = Standby
                         ISSU State = Load Version
                     Operating Mode = Stateful Switchover
                      Current Image = bootflash:new_image
          Pre-ISSU (Original) Image = bootflash:old_image
         Post-ISSU (Targeted) Image = bootflash:new_image

Switch# show issu rollback-timer
        Rollback Process State = Not in progress
      Configured Rollback Time = 60:00

Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# issu set rollback-timer 20
% ISSU state should be [ init ] to set the rollback timer
```

The ISSU Compatibility Matrix

The ISSU Compatibility Matrix contains information about the compatibility of the IOS XE software version currently running on the system, and other versions. The Compatibility Matrix deals with two kinds of information:

- Stored Information, page 1-79
- Negotiated Information, page 1-80

Stored Information

The stored compatibility matrix contains a list of other IOS XE software releases for the Catalyst 4500 platform that are compatible with this release. This information is precomputed and stored inside the IOS XE image.

When an ISSU upgrade is attempted, the software looks up the Stored Compatibility Matrix on the supervisor engine that is running the higher (that is, later) IOS XE version. In this matrix, the software tries to locate the IOS XE version number that is running on the other supervisor engine (that is, the lower or earlier version number). If this information is missing, the ISSU upgrade cannot proceed.

All current IOS XE releases on the Catalyst 4500 platform support Dynamic Image Version Capability (DIVC). This means that the ISSU compatibility for the specified version is dynamically computed, as illustrated with the following example:

```
Switch# show issu comp-matrix stored

Number of Matrices in Table = 1

        (1) Matrix for cat4500e-UNIVERSAL-M(182) - cat4500e-UNIVERSAL-M(182)
        ==========================================
        Start Flag (0xDEADBABE)

                My Image ver:  03.03.01.SG
                Peer Version    Compatibility
                ------------    -------------
                03.02.00.SG             Dynamic(0)
                03.02.01.SG             Dynamic(0)
                03.02.00.XO             Dynamic(0)
                03.02.02.SG             Dynamic(0)
                03.02.03.SG             Dynamic(0)
                03.02.04.SG             Dynamic(0)
                03.03.00.SG             Dynamic(0)
                03.03.01.SG             Comp(3)
```

The above Stored Compatibility Matrix is for IOS XE version 03.03.01.SG.

The "Comp(3)" entry shows that IOS XE version 03.03.01.SG is compatible with this version, and the end result is guaranteed to succeed.

The "Dynamic(0)" entry against IOS XE version 03.02.04.SG means that an ISSU upgrade from or to IOS XE version 03.02.04.SG is e permitted. However, the end result will depend upon the ability of individual software features comprising the two versions to successfully complete the ISSU negotiation.

IOS XE version 03.01.01.SG is not in the list. This means that an ISSU upgrade from that version to this IOS XE version (03.03.01.SG) is not possible.

Negotiated Information

While the Stored compatibility matrix information is used before an ISSU upgrade is attempted, the Negotiated compatibility matrix information pertains to the ISSU state after or during an ISSU upgrade attempt. It contains information about how the different software components comprising the IOS XE images on the two supervisor engines were able to negotiate their states. So, this data is useful for troubleshooting failed ISSU upgrade operations.

To display information about the ISSU compatibility matrix, perform this task:

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch# show issu comp-matrix {negotiated | xml}`	Displays the negotiated ISSU compatibility matrix, either in plain text or in XML form. • **negotiated**—Displays negotiated compatibility matrix information in plain text. • **xml**—Displays negotiated compatibility matrix information in XML form. **Note** These commands display only the data within IOSd process. Use the **show package compatibility** command to display the information for the whole system.
Step 3	`Switch# show package compatibility`	Displays information regarding all client compatibility in the system.

This example shows how to display negotiated information regarding the compatibility matrix:

```
Switch> enable
Switch# show issu comp-matrix negotiated

CardType: WS-C4503-E(182), Uid: 11,  Image Ver: 15.1(2)SGN1.34
Image Name: cat4500e-UNIVERSAL-M

Cid     Eid     Sid     pSid    pUid      Compatibility
=======================================================
2       1       131111  4       1         COMPATIBLE
3       1       65617   7       1         COMPATIBLE
4       1       131085  11      1         COMPATIBLE
5       1       131115  13      1         COMPATIBLE
...
...
7200    1       131105  75      1         COMPATIBLE
7201    1       131151  76      1         COMPATIBLE
7203    1       131127  74      1         COMPATIBLE
7301    1       131137  77      1         COMPATIBLE

Message group summary:
Cid     Eid     GrpId     Sid     pSid    pUid    Nego Result
=============================================================
2       1       1         131111  4       1       Y
3       1       1         65617   7       1       Y
4       1       1         131085  11      1       Y
5       1       1         131115  13      1       Y
```

```
...
...
7200    1       1         131105  75      1       Y
7201    1       1         131151  76      1       Y
7203    1       1         131127  74      1       Y
7301    1       1         131137  77      1       Y

List of Clients:
Cid       Client Name              Base/Non-Base
================================================
2         ISSU Proto client        Base
3         ISSU RF                  Base
4         ISSU CF client           Base
5         ISSU Network RF client   Base
...
...
7200      ISSU Archive Client      Non-Base
7201      ISSU Rollback Client     Non-Base
7203      ISSU Shell Client        Non-Base
7301      ISSU ANCP Client         Non-Base

Switch#
```

This example shows how to display negotiated information regarding non-IOSd clients:

```
Switch# show package compatibility
     PackageName      PeerPackageName                       ModuleName    Compatibility
     -----------      --------------- --------------------------------    -------------
         rp_base              rp_base                              aaa       COMPATIBLE
         rp_base              rp_base                        aaacommon       COMPATIBLE
         rp_base              rp_base                    access_policy       COMPATIBLE
         rp_base              rp_base                         app_sess       COMPATIBLE
         rp_base              rp_base                     app_sess_ios       COMPATIBLE
         rp_base              rp_base                         auth_mgr       COMPATIBLE
......
......
```

License Upgrade on a VSS

When previous license is about to expire, or a new license is to be installed, you need to perform the license update procedure for VSS.

Step 1 Shutdown all non-VSL ports of the VSS standby and perform a "write memory" to save the configuration. When non-VSL ports of the VSS standby are shutdown, the VSS standby is effectively removed from the network.

Step 2 Install a new license on the VSS active. The VSS standby may enter into RPR mode provided the license level is not "good enough" for compatibility with newly installed license on VSS active.

Step 3 Shut down VSL ports on the VSS active.

Caution Shutting down the VSL ports on the VSS active detaches the standby, which might transition to the VSS active. If not, reload the VSS standby and allow it boot as the VSS active.

Note A VSS standby booting as the active does not pose a network problem because all non-VSL ports are shutdown.

Step 4 Install the license on the former VSS standby, the one also functioning as the active.

Note During this time, the VSS active operates without interruption.

Step 5 Ensure that the VSL ports of the VSS standby are not administratively down (At this point, the the VSS standby is also functioning as the active, although all non-VSL ports are shutdown).

Step 6 Reboot the VSS standby.

Step 7 While the VSS standby switch boots, unshut the VSL ports on the active switch. The original VSS standby switch boots as the VSS standby.

Step 8 When the VSS standby achieves SSO-HOT, open all shutdown ports sequentially.

Step 9 Failover the chassis to bring the VSS pair license to the correct level.

This reloads the current VSS active and forces the VSS standby to take over as the VSS active with the new license level.

License install and subsequent VSS formation are now complete.