CHAPTER 4

Configuring Virtual Switching Systems

This chapter describes how to configure a virtual switching system (VSS) for the Catalyst 6500 series switch. Cisco IOS Release 12.2(33)SXH1 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 Release 12.2 publications at this URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.html

Tip For additional information about Cisco Catalyst 6500 series switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

This chapter consists of these sections:

- Understanding Virtual Switching Systems, page 4-1
- VSS Configuration Guidelines and Restrictions, page 4-27
- Configuring a VSS, page 4-29
- Upgrading a VSS, page 4-54

Understanding Virtual Switching Systems

These sections describe a VSS:

- VSS Overview, page 4-2
- VSS Redundancy, page 4-11
- Multichassis EtherChannels, page 4-14

- Packet Handling, page 4-16
- System Monitoring, page 4-20
- Dual-Active Detection, page 4-22
- VSS Initialization, page 4-24
- VSS Configuration Guidelines and Restrictions, page 4-27

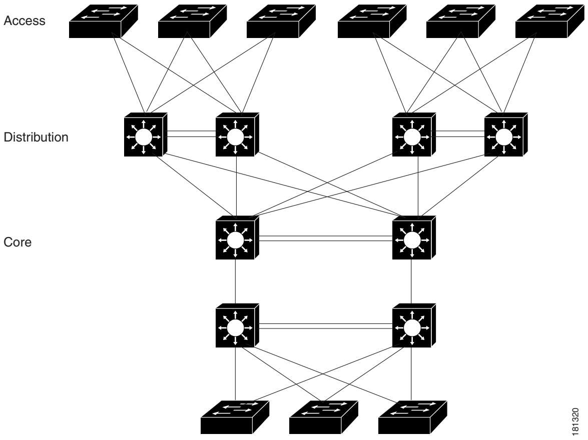
VSS Overview

Network operators increase network reliability by configuring redundant pairs of network devices and links. Figure 4-1 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 6500 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 4-1 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 4-3
- VSS Functionality, page 4-6
- Hardware Requirements, page 4-8
- Understanding VSL Topology, page 4-11

Key Concepts

The VSS incorporates the following key concepts:

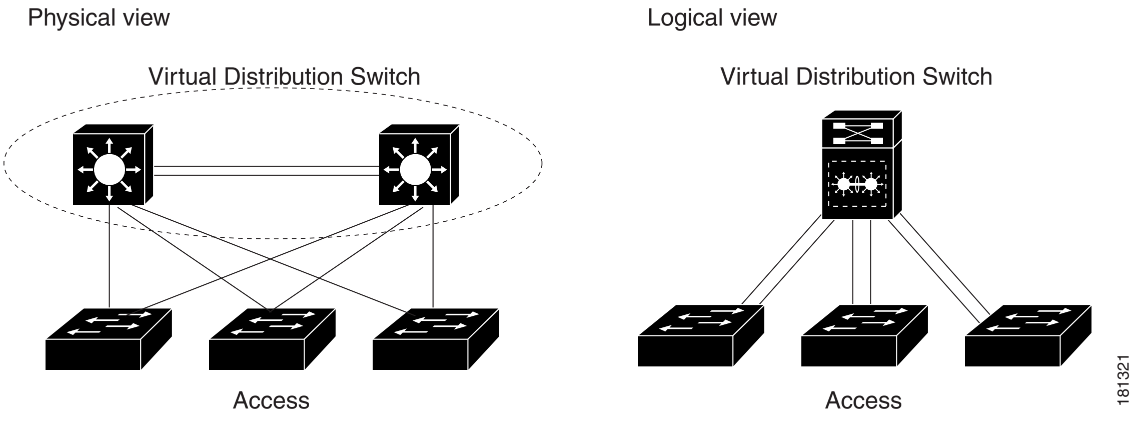
- Virtual Switching System, page 4-3
- VSS Active and VSS Standby Chassis, page 4-3
- Virtual Switch Link, page 4-4
- Multichassis EtherChannel, page 4-5

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 4-2.

An access switch connects to both chassis 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 because the VSS reduces the number of routing peers in the network.

Figure 4-2 VSS in the Distribution Network



VSS Active and VSS Standby Chassis

When you create or restart a VSS, the peer chassis negotiate their roles. One chassis becomes the VSS active chassis, and the other chassis becomes the VSS standby.

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

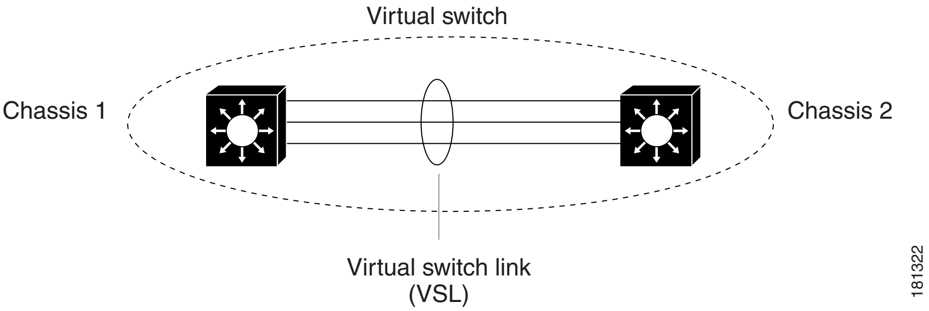
The VSS active and VSS standby chassis perform packet forwarding for ingress data traffic on their locally hosted interfaces. However, the VSS standby chassis sends all control traffic to the VSS active chassis for processing.

Virtual Switch Link

For the two chassis 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 chassis of a VSS, as shown in Figure 4-3. The VSL is implemented as an EtherChannel with up to eight links. The VSL gives control traffic higher priority than data traffic so that control messages are never discarded. Data traffic is load balanced among the VSL links by the EtherChannel load-balancing algorithm.

Figure 4-3 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. When an interface is in restricted mode, only specific configuration commands can be configured on the interface.

The following VSL configuration commands are not removed from the interface when it becomes restricted:

- **mls qos trust cos**
- **mls qos channel-consistency**
- **description**
- **logging event**
- **load-interval**
- **vslp**
- **port-channel port**

When in VSL restricted mode, only these configuration commands are available:

- **channel-group**
- **default**
- **description**
- **exit**
- **load-interval**
- **logging**
- **mls**
- **mls ip**

- **mls ipx**
- **mls netflow**
- **mls rp**
- **mls switching**
- **no**
- **shutdown**

Note The **mls qos** command is not available when a port is in VSL restricted mode.

Multichassis EtherChannel

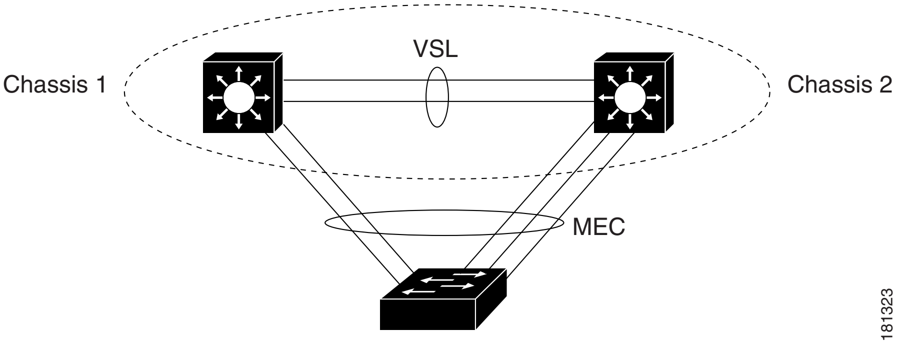
An EtherChannel (also known as a port channel) is a collection of two or more physical links that combine to form one logical link. Layer 2 protocols operate on the EtherChannel as a single logical entity.

A multichassis EtherChannel (MEC) is a port channel that spans the two chassis of a VSS. The access switch views the MEC as a standard port channel. See Figure 4-4.

The VSS supports a maximum of 512 EtherChannels. This limit applies to the combined total of regular EtherChannels and MECs. Because VSL requires two EtherChannel numbers (one for each chassis), there are 510 user-configurable EtherChannels. If an installed service module uses an internal EtherChannel, that EtherChannel will be included in the total.

Note For releases earlier than Cisco IOS Release 12.2(33)SXI, the maximum number of EtherChannels is 128, allowing 126 user-configurable EtherChannels.

Figure 4-4 VSS with MEC



VSS Functionality

The following sections describe the main functionality of a VSS:

- Redundancy and High Availability, page 4-6
- Packet Handling, page 4-6
- System Management, page 4-6
- VSS Quad-Sup Uplink Forwarding, page 4-7
- Interface Naming Convention, page 4-8
- Software Features, page 4-8

Redundancy and High Availability

In a VSS, supervisor engine redundancy operates between the VSS active and VSS standby chassis, using stateful switchover (SSO) and nonstop forwarding (NSF). The peer chassis exchange configuration and state information across the VSL and the VSS standby supervisor engine runs in hot VSS standby mode.

The VSS standby chassis monitors the VSS active chassis using the VSL. If it detects failure, the VSS standby chassis initiates a switchover and takes on the VSS active role. When the failed chassis recovers, it takes on the VSS standby role.

If the VSL fails completely, the VSS standby chassis assumes that the VSS active chassis has failed, and initiates a switchover. After the switchover, if both chassis are VSS active, the dual-active detection feature detects this condition and initiates recovery action. For additional information about dual-active detection, see the "Dual-Active Detection" section on page 4-22.

Packet Handling

The VSS active supervisor engine runs the Layer 2 and Layer 3 protocols and features for the VSS and manages the DFC modules for both chassis.

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

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

Because the VSS standby chassis is actively forwarding traffic, the VSS active supervisor engine distributes updates to the VSS standby supervisor engine PFC and all VSS standby chassis DFCs.

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 chassis. The VSS active supervisor engine uses VSL to send messages to and from local ports on the VSS standby chassis.

The command console on the VSS active supervisor engine is used to control both chassis. In virtual switch mode, the command console on the VSS standby supervisor engine blocks attempts to enter configuration mode.

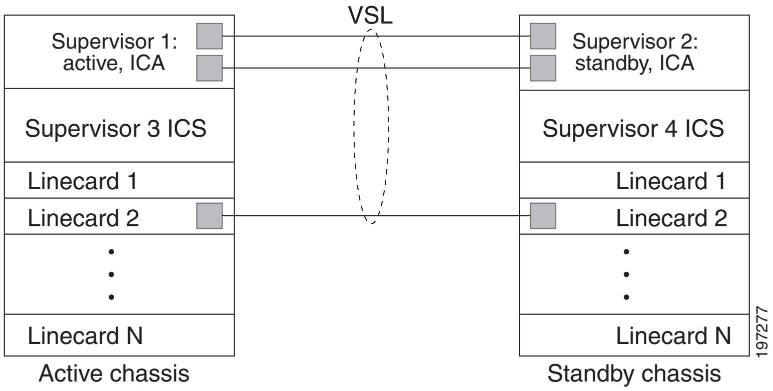
The VSS standby chassis runs a subset of system management tasks. For example, the VSS standby chassis handles its own power management.

VSS Quad-Sup Uplink Forwarding

When you use VSS quad-supervisor uplink forwarding, the in-chassis standby (ICS) supervisor engine acts as a DFC line card. Only one processor, the SP processor, acts as the DFC line card; the RP processor is reset to ROMMON. During the bootup, once the chassis level role is resolved, the ICS downloads the image from the in-chassis active (ICA) supervisor engine. Once the supervisor engine is booted with the image, it will function in the same way as a DFC line card. All applications running in virtual switch (VS) view the in-chassis standby as a DFC line card.

See Figure 4-6 for the various roles that supervisor engines can assume within a quad-supervisor VSS system.

Figure 4-5 Typical VSS Quad-Supervisor Configuration



If your supervisor engine is:

- in-chassis active, it can be VSS active or VSS standby.
- in-chassis standby, it can only be an ICS.
- VSS active, it can only be ICA.
- VSS standby, it can only be ICA.

Quad-supervisor uplink forwarding provides these key features:

- eFSU upgrades— You can upgrade or downgrade your VSS system using ISSU. See “Upgrading a VSS” section on page 4-54 for more information about eFSU upgrades.
- Image version mismatch—Before the bootup, the ICS completes a version check. If there is a version mismatch, the ICS is set to ROMMON. If you want to boot different images on the ICS and ICA. You need to configure the **no switch virtual in-chassis standby bootup version mismatch-check** command. This command is only valid once all four supervisors are running software that supports Quad-supervisor uplink forwarding. If one supervisor is running software that does not support Quad-supervisor uplink forwarding the command will have no effect.
- EARL mode mismatch—If the supervisor engine EARL modes do not match then the supervisor engine is reset to ROMMON. It is recommended that all four supervisor engines run the same EARL Lite or EARL Heavy version.
- VSS RPR switchover—On RPR switchover the ICS will be reset. For more information regarding RPR see “RPR and SSO Redundancy” section on page 4-12.

- In-chassis RPR switchover—ICS supervisor engines in the supervisor engine 1 and supervisor engine 2 positions boot up as RPR-Warm. RPR-Warm is when a supervisor engine acts as a DFC. When a VSS stateful switchover occurs, the supervisor engine is reset to ROMMON and boot ups with the supervisor engine image. You can verify the switchover mode of the supervisor engines by entering the **show module** command.
- VSS stateful switchover—When the in-chassis active supervisor engine crashes, a switchover occurs and the whole chassis reloads (including the ICS) during which the standby supervisor engine takes over as the in-chassis active supervisor engine. A z-switchover operates exactly like a switchover except that the ICS supervisor engine takes priority and is assigned the in-chassis standby supervisor engine. You can initiate a z-switchover by entering the **redundancy force switchover** command on the in-chassis active supervisor engine. You can verify the switchover mode of the supervisor engines by entering the **show module** command.

If you insert a supervisor engine from another system (VS or standalone) in the supervisor engine 1 or supervisor engine 2 position of your existing two supervisor engine VSS system, the supervisor engine does a reset to update the supervisor engine number, and then reboots before going online as a DFC.

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.

Software Features

With some exceptions, the VSS has feature parity with the standalone Catalyst 6500 series switch. Major exceptions include:

- In software releases earlier than Cisco IOS Release 12.2(33)SXI2, the VSS does not support IPv6 unicast or MPLS.
- In software releases earlier than Cisco IOS Release 12.2(33)SXI, port-based QoS and port ACLs (PACLs) are supported only on Layer 2 single-chassis or multichassis EtherChannel (MEC) links. Beginning with Cisco IOS Release 12.2(33)SXI, port-based QoS and PACLs can be applied to any physical port in the VSS, excluding ports in the VSL. PACLs can be applied to no more than 2046 ports in the VSS.
- In software releases earlier than Cisco IOS Release 12.2(33)SXI4, the VSS does not support supervisor engine redundancy within a chassis.
- Starting in Cisco IOS Release 12.2(33)SXI4, the VSS does support supervisor engine redundancy within a chassis.
- In releases earlier than Release 12.2(33) SXH2, the VSS feature and the lawful intercept feature cannot be configured together. (CSCsl77715)

Hardware Requirements

The following sections describe the hardware requirements of a VSS:

- Chassis and Modules, page 4-9
- VSL Hardware Requirements, page 4-9
- PFC, DFC, and CFC Requirements, page 4-10

- Multichassis EtherChannel Requirements, page 4-10
- Service Module Support, page 4-10

Chassis and Modules

Table 4-1 describes the hardware requirements for the VSS chassis and modules.

Table 4-1 VSS Hardware Requirements

Hardware	Count	Requirements
Chassis	2	The VSS is available on chassis that support VS-S720-10G supervisor engines and WS-X6708-10G switching modules. **Note** The two chassis need not be identical.
Supervisor Engines	2	The VSS requires Supervisor Engine 720 with 10-Gigabit Ethernet ports. You must use either two VS-S720-10G-3C or two VS-S720-10G-3CXL supervisor engine modules. The two supervisor engines must match exactly.
Switching Modules	2+	The VSS requires 67xx series switching modules. The VSS does not support classic, CEF256, or dCEF256 switching modules. In virtual switch mode, unsupported switching modules remain powered off.

VSL Hardware Requirements

The VSL EtherChannel supports only 10-Gigabit Ethernet ports. The 10-Gigabit Ethernet port can be located on the supervisor engine module or on one of the following switching modules:

- WS-X6708-10G-3C or WS-X6708-10G-3CXL
- WS-X6716-10G-3C or WS-X6716-10G-3CXL
- WS-X6716-10T-3C or WS-X6716-10T-3CXL

Note

- Using the 10-Gigabit Ethernet ports on a WS-X6716-10G switching module in the VSL EtherChannel requires Cisco IOS Release 12.2(33)SXI or a later release.
- Using the 10-Gigabit Ethernet ports on a WS-X6716-10T switching module in the VSL EtherChannel requires Cisco IOS Release 12.2(33)SXI4 or a later release.

We recommend that you use both of the 10-Gigabit Ethernet ports on the supervisor engines to create the VSL between the two chassis.

You can add additional physical links to the VSL EtherChannel by using the 10-Gigabit Ethernet ports on WS-X6708-10G, WS-X6716-10G, or WS-X6716-10T switching modules.

Note

- When using the 10-Gigabit Ethernet ports on the WS-X6716-10G or WS-X6716-10T switching module as VSL links, you must operate the ports in performance, not oversubscription, mode. If you enter the **no hw-module switch x slot y oversubscription** command to configure non-oversubscription mode (performance mode), then only ports 1, 5, 9, and 13 are configurable; the other ports on the module are disabled.
- Port-groups are independent of each other and one, or more, port-groups can operate in non-oversubscribed (1:1) mode (e.g. for VSL) with the 3 unused ports administratively shutdown, while the others can still operate in oversubscribed (4:1) mode.

PFC, DFC, and CFC Requirements

The VSS supports any 67xx series switching module with CFC hardware.

The VSS supports DFC3C or DFC3CXL hardware and does not support DFC3A/3B/3BXL hardware.

If any switching module in the VSS is provisioned with DFC3C, the whole VSS must operate in PFC3C mode. If a 67xx series switching module with a DFC3A/3B/3BXL is inserted in the chassis of a VSS, the module will remain unpowered, because VSS supports only DFC3C and DFC3CXL.

If the supervisor engines are provisioned with PFC3C, the VSS will automatically operate in 3C mode, even if some of the modules are 3CXL. However, if the supervisor engines are provisioned with PFC3CXL, but some of the modules are 3C, you need to configure the VSS to operate in 3C mode. The **platform hardware vsl pfc mode pfc3c** configuration command sets the system to operate in 3C mode after the next restart. See the “SSO Dependencies” section on page 4-25 for further details about this command.

Multichassis EtherChannel Requirements

Physical links from any 67xx series switching module can be used to implement a Multichassis EtherChannel (MEC).

Service Module Support

VSS mode supports these service modules:

- Network Analysis Modules (NAM):
  - WS-SVC-NAM-1
  - WS-SVC-NAM-2
- Application Control Engines (ACE):
  - ACE10-6500-K9
  - ACE20-MOD-K9
- Intrusion Detection System Services Module (IDSM): WS-SVC-IDSM2-K9
- Wireless Services Module (WiSM): WS-SVC-WISM-1-K9
- Firewall Services Module (FWSM): WS-SVC-FWM-1-K9

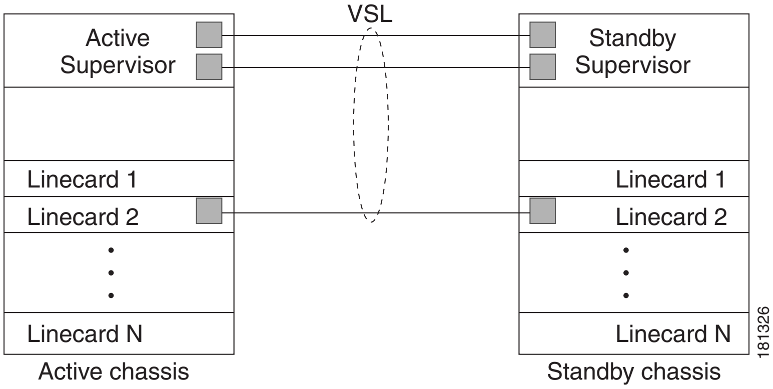
Note Before deploying a service module in VSS mode, check the service module release notes and if necessary, upgrade the service module software.

Understanding VSL Topology

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

We recommend that you configure both of the 10-Gigabit Ethernet ports on the supervisor engines as VSL ports. Optionally, you can also configure the VSL port group to contain switching module 10-Gigabit Ethernet ports. This configuration provides additional VSL capacity. See Figure 4-6 for an example topology.

Figure 4-6 VSL Topology Example



VSS Redundancy

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

- Overview, page 4-11
- RPR and SSO Redundancy, page 4-12
- Failed Chassis Recovery, page 4-13
- VSL Failure, page 4-13
- User Actions, page 4-14

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 chassis and use the VSL to exchange information.
- The VSS active supervisor engine controls both chassis of the VSS. The VSS active supervisor engine runs the Layer 2 and Layer 3 control protocols and manages the switching modules on both chassis.
- The VSS active and VSS standby chassis both 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.
- VSL-related configuration in the two chassis must match.
- PFC mode must match.
- SSO and nonstop forwarding (NSF) must be configured on each chassis.

See the "SSO Dependencies" section on page 4-25 for additional details about the requirements for SSO redundancy on a VSS. See Chapter 6, "Configuring NSF with SSO Supervisor Engine Redundancy" for information about configuring SSO and NSF.

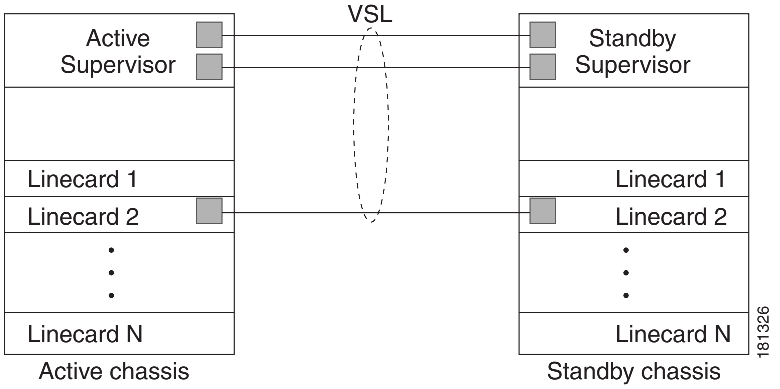
With SSO redundancy, the VSS standby supervisor engine is always ready to assume control following a fault on the VSS active supervisor engine. Configuration, forwarding, and state information are synchronized from the VSS active supervisor engine to the redundant supervisor engine at startup and whenever 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, the VSS will use route processor redundancy (RPR). In RPR mode, the VSS active supervisor engine does not synchronize configuration changes or state information with the VSS standby. The VSS standby supervisor engine is only partially initialized and the switching modules on the VSS standby supervisor are not powered up. If a switchover occurs, the VSS standby supervisor engine completes its initialization and powers up the switching modules. Traffic is disrupted for the normal reboot time of the chassis.

The VSS normally runs stateful switchover (SSO) between the VSS active and VSS standby supervisor engines (see Figure 4-7). The VSS determines the role of each supervisor engine during initialization.

The supervisor engine in the VSS standby chassis 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.

Figure 4-7 Chassis Roles in a VSS



Failed Chassis Recovery

If the VSS active chassis 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 chassis performs recovery action by reloading the supervisor engine.

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

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

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

Note The VSS may experience a brief data path disruption when the switching modules in the VSS standby chassis 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 chassis. 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. Two methods are available to reduce data disruption following an SSO:

- Beginning in Cisco IOS Release 12.2(33)SXH2, you can configure the VSS to activate non-VSL ports in smaller groups over a period of time rather than all ports simultaneously. For information about deferring activation of the ports, see the "Configuring Deferred Port Activation During VSS Standby Recovery" section on page 4-44.
- You can defer the load sharing of the peer switch's MEC member ports during reestablishment of the port connections. See the "Failed Chassis MEC Recovery" section on page 4-16 for details about load share deferral.

VSL Failure

To ensure fast recovery from VSL failures, fast link notification is enabled in virtual switch mode on all port channel members (including VSL ports) whose hardware supports fast link notification.

Note Fast link notification is not compatible with link debounce mechanisms. In virtual switch mode, link debounce is disabled on all port channel members.

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

If only the VSL has failed and the VSS active chassis is still operational, this is a dual-active scenario. The VSS detects that both chassis are operating in VSS active mode and performs recovery action. See the "Dual-Active Detection" section on page 4-22 for additional details about the dual-active scenario.

User Actions

From the VSS active chassis command console, you can initiate a VSS switchover or a reload.

If you enter the **reload** command from the command console, the entire VSS performs a reload.

To reload only the VSS standby chassis, use **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 the VSS standby supervisor engine or to reset both the VSS active and VSS standby supervisor engines, use the **redundancy reload shelf** command.

Multichassis EtherChannels

These sections describe multichassis EtherChannels (MECs):

- Overview, page 4-14
- MEC Failure Scenarios, page 4-15

Overview

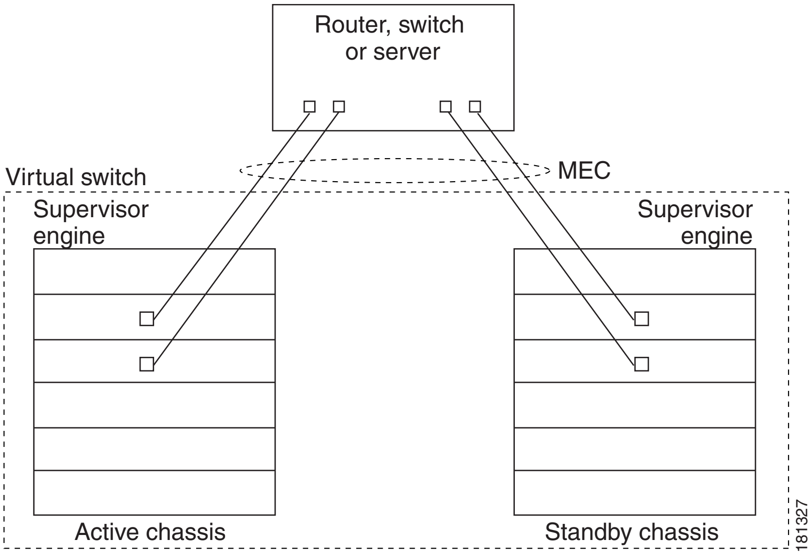
A multichassis EtherChannel is an EtherChannel with ports that terminate on both chassis of the VSS (see Figure 4-8). A VSS MEC can connect to any network element that supports EtherChannel (such as a host, server, router, or switch).

At the VSS, an MEC is an EtherChannel with additional capability: the VSS balances the load across ports in each chassis independently. For example, if traffic enters the VSS active chassis, the VSS will select an MEC link from the VSS active chassis. 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 chassis. PAgP or LACP control packets destined for an MEC link on the VSS standby chassis are sent across VSL.

An MEC can support up to eight VSS active physical links, which can be distributed in any proportion between the VSS active and VSS standby chassis.

Figure 4-8 MEC Topology



MEC Failure Scenarios

We recommend that you configure the MEC with at least one link to each chassis. This configuration conserves VSL bandwidth (traffic egress link is on the same chassis 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 4-15
- All MEC Links to the VSS Active Chassis Fail, page 4-15
- All MEC Links to the VSS Standby Chassis Fail, page 4-16
- All MEC Links Fail, page 4-16
- VSS Standby Chassis Failure, page 4-16
- VSS Active Chassis Failure, page 4-16
- Failed Chassis MEC Recovery, page 4-16

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 Chassis Fail

If all links to the VSS active chassis fail, the MEC becomes a regular EtherChannel with operational links to the VSS standby chassis.

Data traffic terminating on the VSS active chassis reaches the MEC by crossing the VSL to the VSS standby chassis. Control protocols continue to run in the VSS active chassis. Protocol messages reach the MEC by crossing the VSL.

All MEC Links to the VSS Standby Chassis Fail

If all links fail to the VSS standby chassis, the MEC becomes a regular EtherChannel with operational links to the VSS active chassis.

Control protocols continue to run in the VSS active chassis. All control and data traffic from the VSS standby chassis reaches the MEC by crossing the VSL to the VSS active chassis.

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 Chassis Failure

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

VSS Active Chassis Failure

VSS active chassis failure results in a stateful switchover (SSO). See the "VSS Redundancy" section on page 4-11 for details about SSO on a VSS. After the switchover, the MEC is operational on the new VSS active chassis. Connected peer switches detect the link failures (to the failed chassis), and adjust their load-balancing algorithms to use only the links to the new VSS active chassis.

Failed Chassis MEC Recovery

When a failed chassis returns to service as the new VSS standby chassis, protocol messages reestablish the MEC links between the recovered chassis and connected peer switches.

Although the recovered chassis' MEC links are immediately ready to receive unicast traffic from the peer switch, received multicast traffic may be lost for a period of several seconds to several minutes. To reduce this loss, you can configure the port load share deferral feature on MEC port channels of the peer switch. When load share deferral is configured, the peer's deferred MEC port channels will establish with an initial load share of 0. During the configured deferral interval, the peer's deferred port channels are capable of receiving data and control traffic, and of sending control traffic, but are unable to forward data traffic to the VSS. See the "Configuring Port Load Share Deferral on the Peer Switch" section on page 4-45 for details about configuring port load share deferral.

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 DFC modules for both chassis.

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

Both chassis 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 4-17
- Layer 2 Protocols, page 4-17
- Layer 3 Protocols, page 4-18
- SPAN, page 4-20

Traffic on the VSL

The VSL carries data traffic and in-band control traffic between the two chassis. All frames forwarded over the VSL link are encapsulated with a special 32-byte header, which provides information for the VSS to forward the packet on the peer chassis.

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

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 chassis.
- The packet destination is on the peer chassis, such as the following examples:
  - Traffic within a VLAN where the known destination interface is on the peer chassis.
  - Traffic that is replicated for a multicast group and the multicast receivers are on the peer chassis.
  - The known unicast destination MAC address is on the peer chassis.
  - The packet is a MAC notification frame destined for a port on the peer chassis.

VSL also transports system data, such as NetFlow export data and SNMP data, from the VSS standby chassis 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 chassis), the VSS transmits packets to the access switch using a link on the same chassis 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 chassis. Protocol messages that are transmitted and received on the VSS standby chassis switching modules must traverse the VSL to reach the VSS active supervisor engine.

The following sections describe Layer 2 protocols for a VSS:

- Spanning Tree Protocol, page 4-18
- Virtual Trunk Protocol, page 4-18
- EtherChannel Control Protocols, page 4-18
- Multicast Protocols, page 4-18

Spanning Tree Protocol

The VSS active chassis runs Spanning Tree Protocol (STP). The VSS standby chassis redirects STP BPDUs across the VSL to the VSS active chassis.

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.

Virtual Trunk Protocol

Virtual Trunk Protocol (VTP) uses the IP address of the switch and local current time for version control in advertisements. After a switchover, VTP uses the IP address of the newly VSS active chassis.

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 to use.

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 4-22.

Multicast Protocols

In Release 12.2(33)SXI4 and later releases, fast-redirect optimization makes multicast traffic redirection between inter-chassis or intra-chassis line cards faster for Layer 2 trunk multichassis EtherChannel or distributed EtherChannel in case of member port link failure and recovery. This operation occurs mainly when a member port link goes down (port leaves the EtherChannel) and when the member port link goes up (port joins or rejoins the EtherChannel). Fast-redirect does not take effect when you add or remove a member port due to a configuration change or during system boot up.

Layer 3 Protocols

The MSFC on the VSS active supervisor engine runs the Layer 3 protocols and features for the VSS. Both chassis perform packet forwarding for ingress traffic on their interfaces. If possible, ingress traffic is forwarded to an outgoing interface on the same chassis, to minimize data traffic that must traverse the VSL.

Because the VSS standby chassis is actively forwarding traffic, the VSS active supervisor engine distributes updates to the VSS standby supervisor engine PFC and all VSS standby chassis DFCs.

The following sections describe Layer 3 protocols for a VSS:

- IPv4, page 4-18
- IPv6 and MPLS, page 4-19
- IPv4 Multicast, page 4-19
- Software Features, page 4-20

IPv4

The supervisor engine on the VSS active chassis runs the IPv4 routing protocols and performs any required software forwarding.

Routing updates received on the VSS standby chassis are redirected to the VSS active chassis across the VSL.

Hardware forwarding is distributed across all DFCs on the VSS. The supervisor engine on the VSS active chassis sends FIB updates to all local DFCs, remote DFCs, and the VSS standby supervisor engine PFC.

All hardware routing uses the router MAC address assigned by the VSS active supervisor engine. After a switchover, the original MAC address is still used.

The supervisor engine on the VSS active chassis 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) are the same as in standalone mode. For additional information about NSF requirements, refer to the *Catalyst 6500 Series Switch Cisco IOS Configuration Guide*, Release 12.2SX.

From a routing peer perspective, EtherChannels remain operational during a switchover (only the links to the failed chassis are down).

The VSS implements path filtering by storing only local paths (paths that do not traverse the VSL) in the FIB entries. Therefore, IP forwarding performs load sharing among the local paths. If no local paths to a given destination are available, the VSS updates the FIB entry to include remote paths (reachable by traversing the VSL).

IPv6 and MPLS

In Cisco IOS Release 12.2(33)SXI2 and later releases, the VSS supports IPv6 unicast and MPLS.

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 chassis.

The VSS active supervisor engine sends IGMP and PIM protocol packets to the VSS standby supervisor engine in order to maintain Layer 2 information for stateful switchover (SSO).

The VSS active supervisor engine distributes multicast FIB and adjacency table updates to the VSS standby supervisor engine and switching module DFCs.

For Layer 3 multicast in the VSS, learned multicast routes are stored in hardware in the VSS standby supervisor engine. After a switchover, multicast forwarding continues, using the existing hardware entries.

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).

In virtual switch mode, the VSS active chassis does not program the multicast expansion table (MET) on the VSS standby chassis. The VSS standby supervisor engine programs the outgoing interface hardware entries for all local multicast receivers

If all switching modules on the VSS active chassis and VSS standby chassis are egress capable, the multicast replication mode is set to egress mode; otherwise, the mode is set to ingress mode.

In egress mode, replication is distributed to DFCs that have ports in outgoing VLANs for a particular flow. In ingress mode, replication for all outgoing VLANs is done on the ingress DFC.

For packets traversing VSL, all Layer 3 multicast replication occurs on the ingress chassis. If there are multiple receivers on the egress chassis, replicated packets are forwarded over the VSL.

Software Features

Software features run only on the VSS active supervisor engine. Incoming packets to the VSS standby chassis that require software processing are sent across the VSL.

For features supported in hardware, the ACL configuration is sent to the TCAM manager on the VSS active supervisor engine, the VSS standby supervisor engine, and all DFCs.

SPAN

The VSS supports all SPAN features for non-VSL interfaces. The VSS supports SPAN features on VSL interfaces with the following limitations:

- If the VSL is configured as a local SPAN source, the SPAN destination interface must be on the same chassis as the source interface.
- VSL cannot be configured as a SPAN destination.
- VSL cannot be configured as a traffic source of RSPAN, ERSPAN, or egress-only SPAN.

The number of SPAN sessions available to a VSS is the same as for a single chassis running in standalone mode.

System Monitoring

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

- Power Management, page 4-20
- Environmental Monitoring, page 4-20
- File System Access, page 4-21
- VSL Diagnostics, page 4-21
- Service Modules, page 4-21
- Network Management, page 4-22

Power Management

From the VSS active chassis, you can control power-related functions for the VSS standby chassis. For example, use the **power enable switch** command to control power to the modules and slots on the VSS standby chassis. Use the **show power switch** command to see the current power settings and status.

Environmental Monitoring

Environmental monitoring runs on both supervisor engines. The VSS standby chassis reports notifications to the VSS active supervisor engine. The VSS active chassis gathers log messages for both chassis. The VSS active chassis synchronizes the calendar and system clock to the VSS standby chassis.

File System Access

You can access file systems of both chassis from the VSS active chassis. Prefix the device name with the switch number and slot number to access directories on the VSS standby chassis. For example, the command **dir sw2-slot6-disk0:** lists the contents of disk0 on the VSS standby chassis (assuming switch 2 is the VSS standby chassis). You can access the VSS standby chassis file system only when VSL is operational.

VSL Diagnostics

You can use the **diagnostic schedule** and **diagnostic start** commands on a VSS. In virtual switch mode, these commands require an additional parameter, which specifies the chassis to apply the command.

When you configure a VSL port on a switching module or a supervisor engine module, the diagnostics suite incorporates additional tests for the VSL ports.

Use the **show diagnostic content** command to display the diagnostics test suite for a module.

The following VSL-specific diagnostics tests are available on WS-X6708-10G switching modules with VSL ports. These tests are disruptive:

- TestVslBridgeLink
- TestVslLocalLoopback

The following VSL-specific diagnostics tests are available on a Supervisor Engine 720-10GE with VSL ports. These tests are disruptive:

- TestVSActiveToStandbyLoopback
- TestVslBridgeLink
- TestVslLocalLoopback

The following VSL-specific diagnostics test is available for VSL ports on the switching module or the supervisor engine. This test is not disruptive:

- TestVslStatus

See the "ViSN Tests" section on page B-47.

Service Modules

The following system monitoring and system management guidelines apply to service modules supported by the VSS:

- The supervisor engine in the same chassis as the service module controls the powering up of the service module. After the service module is online, you initiate a session from the VSS active supervisor engine to configure and maintain the service module.
- Use the **session** command to connect to the service module. If the service module is in the VSS standby chassis, the session runs over the VSL.
- The VSS active chassis performs the graceful shutdown of the service module, even if the service module is in the VSS standby chassis.

Network Management

The following sections describe network management for a VSS:

- Telnet over SSH Sessions and the Web Browser User Interface, page 4-22
- SNMP, page 4-22
- Command Console, page 4-22

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 — Chassis Role and Uptime
- 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

Connect console cables to both supervisor engine console ports. You can only use configuration mode in the console for the VSS active supervisor engine.

The console on the VSS standby chassis will indicate that chassis 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 chassis console.

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

```
Router-stdby> show switch virtual

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

Dual-Active Detection

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

If the original VSS active chassis is still operational, both chassis are now VSS active. This situation is called a *dual-active scenario*. A dual-active scenario can have adverse affects on network stability, because both chassis 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 these three methods for detecting a dual-active scenario:

- Enhanced PAgP—Uses PAgP messaging over the MEC links to communicate between the two chassis through a neighbor switch. Enhanced PAgP is faster than IP BFD, but requires a neighbor switch that supports the PAgP enhancements.
- IP Bidirectional Forwarding Detection (BFD)—Uses BFD messaging over a backup Ethernet connection. IP BFD uses a direct connection between the two chassis and does not require support from a neighbor switch.
- dual-active fast-hello—Uses special hello messages over a backup Ethernet connection. Dual-active fast-hello is faster than IP BFD and does not require support from a neighbor switch. This method is available only in Cisco IOS Release 12.2(33)SXI and later releases,

You can configure all three detection methods to be VSS active at the same time.

For line redundancy, we recommend dedicating at least two ports per switch for dual-active detection. For module redundancy, the two ports can be on different switching modules in each chassis, and should be on different modules than the VSL links, if feasible.

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

- Dual-Active Detection Using Enhanced PAgP, page 4-23
- Dual-Active Detection Using IP BFD, page 4-24
- Dual-Active Detection Using Dual-Active Fast Hello Packets, page 4-24
- Recovery Actions, page 4-24

Dual-Active Detection Using Enhanced PAgP

If a VSS MEC terminates on a Cisco switch, you can run the port aggregation protocol (PAgP) on the MEC. If enhanced PAgP is running on an MEC between the VSS and another switch running Release 12.2(33)SXH1 or a later release, the VSS can use enhanced PAgP to detect a dual-active scenario.

The MEC must have at least one port on each chassis of the VSS. In VSS mode, PAgP messages include a new type length value (TLV) that contains the ID of the VSS active switch. Only switches in VSS mode send the new TLV.

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

If the formerly VSS active chassis is still operational, it detects the dual-active scenario because the VSS active ID in the PAgP messages changes. This chassis initiates recovery actions as described in the "Recovery Actions" section on page 4-24.

Dual-Active Detection Using IP BFD

To use the IP BFD detection method, you must provision a direct Ethernet connection between the two switches. Regular Layer 3 ping will not function correctly on this connection, as both chassis have the same IP address. The VSS instead uses the Bidirectional Forwarding Detection (BFD) protocol.

If the VSL fails, both chassis create BFD neighbors, and try to establish adjacency. If the original VSS active chassis receives an adjacency message, it realizes that this is a dual-active scenario, and initiates recovery actions as described in the "Recovery Actions" section on page 4-24.

Note If Flex Links are configured on the VSS, we recommend using the PAgP detection method. Do not configure Flex Links and BFD dual-active detection on the same VSS.

Dual-Active Detection Using Dual-Active Fast Hello Packets

Cisco IOS Release 12.2(33)SXI and later releases support the dual-active fast hello method.

To use the dual-active fast hello packet detection method, you must provision a direct Ethernet connection between the two VSS chassis. You can dedicate up to four non-VSL links for this purpose.

The two chassis periodically exchange special Layer 2 dual-active hello messages containing information about the switch state. If the VSL fails and a dual-active scenario occurs, each switch recognizes from the peer's messages that there is a dual-active scenario and initiates recovery actions as described in the "Recovery Actions" section on page 4-24. If a switch does not receive an expected dual-active fast hello message from the peer before the timer expires, the switch assumes that the link is no longer capable of dual-active detection.

Recovery Actions

An VSS active chassis that detects a dual-active condition shuts down all of its non-VSL interfaces (except interfaces configured to be excluded from shutdown) to remove itself from the network, and waits in recovery mode until the VSL links have recovered. You might need to physically repair the VSL failure. When the shut down chassis detects that VSL is operational again, the chassis reloads and returns to service as the VSS standby chassis.

Loopback interfaces are also shut down in recovery mode. Do not configure loopback interfaces while in recovery mode, because any new loopback interfaces configured in recovery mode will not be shut down.

Note If the running configuration of the chassis in recovery mode has been changed without saving, the chassis will not automatically reload. In this situation, you must save the running configuration and then reload manually.

VSS Initialization

A VSS is formed when the two chassis and the VSL link between them become operational. The peer chassis communicate over the VSL to negotiate the chassis roles.

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

VSS initialization is described in the following sections:

- Virtual Switch Link Protocol, page 4-25
- SSO Dependencies, page 4-25
- Initialization Procedure, page 4-26

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 chassis use Role Resolution Protocol (RRP) to negotiate the role (VSS active or VSS standby) for each chassis.
- Link Management Protocol—The Link Management Protocol (LMP) runs on all VSL links, and exchanges information required to establish communication between the two chassis. LMP identifies and rejects any unidirectional links. If LMP flags a unidirectional link, the chassis 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—Both supervisor engine modules on the VSS must be running the identical software version.
- VSL configuration consistency—During the startup sequence, the VSS standby chassis sends virtual switch information from the startup-config file to the VSS active chassis. The VSS active chassis ensures that the following information matches correctly on both chassis:
  - Switch virtual domain
  - Switch virtual node
  - Switch priority
  - VSL port channel: switch virtual link identifier
  - VSL ports: channel-group number, shutdown, total number of VSL ports
  - Power redundancy-mode
  - Power enable on VSL modules

  If the VSS detects a mismatch, it prints out an error message on the VSS active chassis console and the VSS standby chassis comes up in RPR mode.

  After you correct the configuration file, save the file by entering the **copy running-config startup-config** command on the VSS active chassis, and then restart the VSS standby chassis.
- PFC mode check—If both supervisor engines are provisioned with PFC3C, the VSS will automatically operate in PFC3C mode, even if some of the switching modules are equipped with DFC3CXLs.

  However, if the supervisor engines are provisioned with PFC3CXL and there is a mixture of DFC3C and DFC3CXL switching modules, the system PFC mode will depend on how the 3C and 3CXL switching modules are distributed between the two chassis.

Each chassis in the VSS determines its system PFC mode. If the supervisor engine of a given chassis is provisioned with PFC3CXL and all the switching modules in the chassis are provisioned with DFC3CXL, the PFC mode for the chassis is PFC3CXL. However, if any of the switching modules is provisioned with DFC3C, the chassis PFC mode will be set to PFC3C. If there is a mismatch between the PFC modes of two chassis, the VSS will come up in RPR mode instead of SSO mode. You can prevent this situation by using the **platform hardware vsl pfc mode pfc3c** command to force the VSS to operate in PFC3C mode after the next reload.

- SSO and NSF enabled

  SSO and NSF must be configured and enabled on both chassis. For detailed information on configuring and verifying SSO and NSF, see Chapter 6, “Configuring NSF with SSO Supervisor Engine Redundancy.”

If these conditions are not met, the VSS operates in RPR redundancy mode. For a description of SSO and RPR, see the “VSS Redundancy” section on page 4-11.

Initialization Procedure

The following sections describe the VSS initialization procedure:

- VSL Initialization, page 4-26
- System Initialization, page 4-26
- VSL Down, page 4-27

VSL Initialization

A VSS is formed when the two chassis and the VSL link between them become operational. Because both chassis 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 chassis communicate over VSL to negotiate their roles (VSS active or VSS standby).
3. The VSS active chassis completes the boot sequence, including the consistency check described in the “SSO Dependencies” section on page 4-25.
4. If the consistency check completed successfully, the VSS standby chassis comes up in SSO VSS standby mode. If the consistency check failed, the VSS standby chassis comes up in RPR mode.
5. The VSS active chassis synchronizes configuration and application data to the VSS standby chassis.

System Initialization

If you boot both chassis simultaneously, the VSL ports become VSS active, and the chassis will come up as VSS active and VSS standby. If priority is configured, the higher priority switch becomes active.

If you boot up only one chassis, the VSL ports remain inactive, and the chassis comes up as VSS active. When you subsequently boot up the other chassis, the VSL links become active, and the new chassis comes up as VSS standby.

VSL Down

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

One of the chassis becomes VSS active and the other chassis initiates recovery from the dual-active scenario. For further information, see the "Configuring Dual-Active Detection" section on page 4-45.

VSS Configuration Guidelines and Restrictions

The following sections describe restrictions and guidelines for VSS configuration:

General VSS Restrictions and Guidelines

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

- The VSS configurations in the startup-config file must match on both chassis.
- If you configure a new value for switch priority, the change takes effect only after you save the configuration file and perform a restart.
- Enable the out-of-band MAC address table synchronization among DFC-equipped switching modules by entering the **mac-address-table synchronize** command.

VSL Restrictions and Guidelines

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

- 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.
- The **no mls qos channel-consistency** command is automatically applied when you configure the VSL. Do not remove this command.
- VSL ports cannot be Mini Protocol Analyzer sources (the **monitor ... capture** command). Monitor capture sessions cannot be started if a source is the VSL on the port channel of the standby switch. The following message is displayed when a remote VSL port channel on the standby switch is specified and you attempt to start the monitor capture:

  ```
  % remote VSL port is not allowed as capture source
  ```

  The following message is displayed when a scheduled monitor capture start fails because a source is a remote VSL port channel:

  ```
  Packet capture session 1 failed to start. A source port is a remote VSL.
  ```

Multichassis EtherChannel Restrictions and Guidelines

When configuring MECs, note the following guidelines and restrictions:

- All links in an MEC must terminate locally on the VSS active or VSS standby chassis of the same virtual domain.
- For an MEC using the LACP control protocol, the *minlinks* command argument defines the minimum number of physical links in each chassis for the MEC to be operational.
- For an MEC using the LACP control protocol, the *maxbundle* command argument defines the maximum number of links in the MEC across the whole VSS.
- MEC supports LACP 1:1 redundancy. For additional information about LACP 1:1 redundancy, refer to the "Understanding LACP 1:1 Redundancy" section on page 19-5.
- An MEC can be connected to another MEC in a different VSS domain.

Dual-Active Detection Restrictions and Guidelines

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

- If Flex Links are configured on the VSS, use PAgP dual-active detection.
- Do not configure Flex Links and BFD dual-active detection on the same VSS.
- For dual-active detection link redundancy, configure at least two ports per switch for dual-active detection. For module redundancy, the two ports can be on different switching modules in each chassis, and should be on different modules than the VSL, if feasible.
- When you configure dual-active fast hello mode, all existing configurations are removed automatically from the interface except for these commands:
  - **description**
  - **logging event**
  - **load-interval**
  - **rcv-queue cos-map**
  - **rcv-queue queue-limit**
  - **rcv-queue random-detect**
  - **rcv-queue threshold**
  - **wrr-queue bandwidth**
  - **wrr-queue cos-map**
  - **wrr-queue queue-limit**
  - **wrr-queue random-detect**
  - **wrr-queue threshold**
  - **priority-queue cos-map**
- Only these configuration commands are available on dual-active detection fast hello ports:
  - **default**
  - **description**
  - **dual-active**

  - **exit**
  - **load-interval**
  - **logging**
  - **no**
  - **shutdown**
- ASIC-specific QoS commands are not configurable on dual-active detection fast hello ports directly, but are allowed to remain on the fast hello port if the commands were configured on another non-fast hello port in that same ASIC group. For a list of these commands, see the "PFC QoS Configuration Guidelines and Restrictions" section on page 43-52.

Service Module Restrictions and Guidelines

When configuring service modules in a VSS, note the following guidelines and restrictions:

- When configuring and attaching VLAN groups to a service module interface in a VSS, use the **switch** {**1** | **2**} command keyword. For example, the **firewall vlan-group** command becomes the **firewall switch** *num* **slot** *slot* **vlan-group** command.
- When upgrading the software image of a service module in a VSS, use the **switch** {**1** | **2**} command keyword.
- EtherChannel load balancing (ECLB) is not supported between an IDSM-2 in the VSS active chassis and an IDSM-2 in the VSS standby chassis.
- A switchover between two service modules in separate chassis of a VSS is considered an intrachassis switchover.

Note For detailed instructions, restrictions, and guidelines for a service module in a VSS, see the configuration guide and command reference for the service module.

Configuring a VSS

These sections describe how to configure a VSS:

Converting to a VSS

By default, the Catalyst 6500 series switch is configured to operate in standalone mode (the switch is a single chassis). 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 chassis will revert to the default configuration.

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

- Save the standalone configuration files.
- Configure SSO and NSF on each chassis.
- Configure each chassis as a VSS.
- Convert to a VSS.
- Configure the peer VSL information.

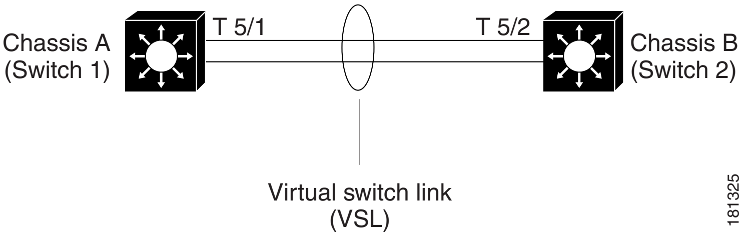
In virtual switch mode, both chassis use the same configuration file. When you make configuration changes on the VSS active chassis, these changes are automatically propagated to the VSS standby chassis.

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

- Backing Up the Standalone Configuration, page 4-31
- Configuring SSO and NSF, page 4-31
- Assigning Virtual Switch Domain and Switch Numbers, page 4-32
- Configuring VSL Port Channel and Ports, page 4-33
- Converting the Chassis to Virtual Switch Mode, page 4-34
- (Optional) Configuring VSS Standby Chassis Modules, page 4-35

In the procedures that follow, the example commands assume the configuration shown in Figure 4-9.

Figure 4-9 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.

Backing Up the Standalone Configuration

Save the configuration files for both chassis 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.

Perform the following task on Switch 2:

	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 must be configured and enabled on both chassis. On Switch 1, perform this task:

	Command	Purpose
Step 1	`Switch-1(config)# redundancy`	Enters redundancy configuration mode.
Step 2	`Switch-1(config-red)# mode sso`	Configures SSO. When this command is entered, the redundant supervisor engine is reloaded and begins to work in SSO mode.
Step 3	`Switch-1(config-red)# exit`	Exits redundancy configuration mode.
Step 4	`Switch-1(config)# router routing_protocol processID`	Enables routing, which places the router in router configuration mode.
Step 5	`Switch-1(config-router)# nsf`	Enables NSF operations for the routing protocol.
Step 6	`Switch-1(config-router)# end`	Exits to privileged EXEC mode.
Step 7	`Switch-1# show running-config`	Verifies that SSO and NSF are configured and enabled.
Step 8	`Switch-1# show redundancy states`	Displays the operating redundancy mode.

Perform the following task on Switch 2:

	Command	Purpose
Step 1	`Switch-2(config)# redundancy`	Enters redundancy configuration mode.
Step 2	`Switch-2(config-red)# mode sso`	Configures SSO. When this command is entered, the redundant supervisor engine is reloaded and begins to work in SSO mode.

	Command	Purpose
Step 3	`Switch-2(config-red)# exit`	Exits redundancy configuration mode.
Step 4	`Switch-2(config)# router routing_protocol processID`	Enables routing, which places the router in router configuration mode.
Step 5	`Switch-2(config-router)# nsf`	Enables NSF operations for the routing protocol.
Step 6	`Switch-2(config-router)# end`	Exits to privileged EXEC mode.
Step 7	`Switch-2# show running-config`	Verifies that SSO and NSF are configured and enabled.
Step 8	`Switch-2# show redundancy states`	Displays the operating redundancy mode.

For detailed information on configuring and verifying SSO and NSF, see Chapter 6, "Configuring NSF with SSO Supervisor Engine Redundancy."

Assigning Virtual Switch Domain and Switch Numbers

You must configure the same virtual switch domain number on both chassis 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 chassis to be switch number 1 and the other chassis to be switch number 2.

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

	Command	Purpose
Step 1	`Switch-1(config)# switch virtual domain 100`	Configures the virtual switch domain on Chassis A.
Step 2	`Switch-1(config-vs-domain)# switch 1`	Configures Chassis 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 Chassis B.
Step 2	`Switch-2(config-vs-domain)# switch 2`	Configures Chassis 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 chassis 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 chassis. During the conversion, the VSS configures both port channels on the VSS active chassis. If the VSS standby chassis 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 chassis.

Check the port channel number by using 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:

	Command	Purpose
Step 1	`Switch-1(config)#` **`interface port-channel 10`**	Configures port channel 10 on Switch 1.
Step 2	`Switch-1(config-if)#` **`switch virtual link 1`**	Associates Switch 1 as owner of port channel 10.
Step 3	`Switch-1(config-if)#` **`no shutdown`**	Activates the port channel.
Step 4	`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-2(config-if)#` **`switch virtual link 2`**	Associates Switch 2 as owner of port channel 20.
Step 3	`Switch-2(config-if)#` **`no shutdown`**	Activates the port channel.
Step 4	`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.
Step 3	`witch-1(config-if)#` **`no shutdown`**	Activates the port.

Perform the following task on Switch 2:

	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.
Step 3	`Switch-2(config-if)# no shutdown`	Activates the port.

Converting the Chassis to Virtual Switch Mode

Conversion to virtual switch mode requires a restart for both chassis. 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 on the RP. This file is assigned a default name, but you are also prompted to override the default name if you want to change it.

Prior to the conversion, ensure that the PFC operating mode matches on both chassis. If they do not match, VSS comes up in RPR redundancy mode. Enter the **show platform hardware pfc mode** command on each chassis to display the current PFC mode. If only one of the chassis is in PFC3CXL mode, you can configure it to use PFC3C mode with the **platform hardware vsl pfc mode pfc3c** command.

To verify the PFC operating mode, perform this task:

	Command	Purpose
Step 1	`Switch-1# show platform hardware pfc mode`	Ensures that the PFC operating mode matches on both chassis, to ensure that the VSS comes up in SSO redundancy mode.
Step 2	`Switch-2# show platform hardware pfc mode`	Ensures that the PFC operating mode matches on both chassis, to ensure that the VSS comes up in SSO redundancy mode.
Step 3	`Switch-1(config)# platform hardware vsl pfc mode pfc3c`	(Optional) Sets the PFC operating mode to PFC3C on Chassis A.
Step 4	`Switch-2(config)# platform hardware vsl pfc mode pfc3c`	(Optional) Sets the PFC operating mode to PFC3C on Chassis B.

To convert Chassis 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 RP bootflash.

To convert Chassis 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 RP 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 chassis reboots. After the reboot, the chassis is in virtual switch mode, so you must specify interfaces with three identifiers (switch/module/port).

(Optional) Configuring VSS Standby Chassis Modules

After the reboot, each chassis contains the module provisioning for its own slots. In addition, the modules from the VSS standby chassis have been automatically provisioned on the VSS active chassis with default configuration.

Configurations for the VSS standby chassis modules are restored to their default settings (for example, no IP addresses).

You can view the module provisioning information in the configuration file, by entering the **show startup-config** command (after you have saved the configuration).

Note Do not delete or modify this section of the configuration file. In Cisco IOS Release 12.2(33)SXI and later releases, you can no longer add module provisioning entries using the **module provision** CLI command. When a module is not present, the provisioning entry for that module can be cleared using the **no slot** command with the **module provision** CLI command. Note that the VSS setup does not support the **module clear-config** command.

The following example shows the module provisioning information from a configuration file:

```
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
```

Displaying VSS Information

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

Command	Purpose
`Router# show switch virtual`	Displays the virtual switch domain number, and the switch number and role for each of the chassis.
`Router# show switch virtual role`	Displays the role, switch number, and priority for each of the chassis in the VSS.
`Router# show switch virtual link`	Displays the status of the VSL.

The following example shows the information output from these commands:

```
Router# show switch virtual
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



Router# 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(100)  ACTIVE   0      0
REMOTE   2     UP      FALSE(N)   100(100)  STANDBY  8158   1991

In dual-active recovery mode: No

Router# show switch virtual link
VSL Status: UP
VSL Uptime: 4 hours, 26 minutes
VSL SCP Ping: Pass OK
VSL ICC (Ping): Pass
VSL Control Link: Te 1/5/1
```

Converting a VSS to Standalone Chassis

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

- Copying the VSS Configuration to a Backup File, page 4-37
- Converting the VSS Active Chassis to Standalone, page 4-37
- Converting the Peer Chassis to Standalone, page 4-37

Copying the VSS Configuration to a Backup File

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

	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 disk0:vs-startup-config`	Copies the startup configuration to a backup file.

Converting the VSS Active Chassis to Standalone

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

The VSS standby chassis of the VSS becomes VSS active. VSL links on this chassis are down because the peer is no longer available.

To convert the VSS active chassis to standalone mode, perform this task on the VSS active chassis:

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**.

Converting the Peer Chassis to Standalone

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

To convert the peer chassis to standalone, perform this task on the VSS standby chassis:

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 4-38
- Configuring PFC Mode, page 4-39
- Configuring PFC Mode, page 4-39
- Configuring a VSL, page 4-40
- Displaying VSL Information, page 4-40
- Configuring VSL QoS, page 4-41
- Subcommands for VSL Port Channels, page 4-42
- Subcommands for VSL Ports, page 4-42
- Configuring the Router MAC Address Assignment, page 4-43

Configuring VSL Switch Priority

To configure the switch priority, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`switch virtual domain 100`**	Enters configuration mode for the virtual switch domain.
Step 2	`Router(config-vs-domain)#` **`switch`** `[`**`1`** `|` **`2`**`]` **`priority`** `[`*`priority_num`*`]`	Configures the priority for the chassis. 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. Enter 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 priority value of 100. The new value takes effect after you save the configuration and perform a reload.
Step 3	`Router#` **`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 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:

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

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

```
Router# 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 PFC Mode

If you have a mixture of DFC3C and DFC3CXL switching modules in the VSS, set the PFC mode by performing this task:

Command	Purpose
`Router(config)# platform hardware vsl pfc mode pfc3c`	Sets the PFC configuration mode for the VSS to PFC3C. **Note** This command requires a system reload before it takes effect.
`Router# show platform hardware pfc mode`	Displays the current settings for the PFC mode.

This example shows how to set the PFC configuration mode for the VSS to PFC3C. You can wait until the next maintenance window to perform the **reload** command.

```
Router(config)# platform hardware vsl pfc mode pfc3c
Router(config)# end
Router# reload
```

If all the supervisor engines and switching modules in the VSS are 3CXL, the following warning is displayed if you set the PFC mode to PFC3C:

```
Router(config)# platform hardware vsl pfc mode pfc3c
PFC Preferred Mode: PFC3CXL. The discrepancy between Operating Mode and
Preferred Mode could be due to PFC mode config. Your System has all PFC3XL modules.
Remove ' platform hardware vsl pfc mode pfc3c ' from global config.
```

This example shows how to display the operating and configured PFC modes:

```
Router# show platform hardware pfc mode
PFC operating mode : PFC3C
Configured PFC operating mode : PFC3C
```

Configuring a VSL

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

	Command	Purpose
Step 1	`Router(config)#` **`interface port-channel`** *`channel_num`*	Enters configuration mode for the specified port channel.
Step 2	`Router(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 chassis 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
Switch-1(config)# interface tenGigabitEthernet 5/1
Switch-1(config-if)# channel-group 10 mode on
Switch-1(config-if)# no shutdown

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

Displaying VSL Information

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

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

This example shows how to display VSL information:

```
Router# show switch virtual link
VSL Status : UP
VSL Uptime : 1 day, 3 hours, 39 minutes
VSL SCP Ping : Pass
VSL ICC Ping : Pass
VSL Control Link : Te 1/5/1

Router# show switch virtual link port-channel
VSL Port Channel Information

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)

Router# show switch virtual link port
VSL Link Info          : Configured: 2 Operational: 1


                          Peer         Peer    Peer
Interface   State         MAC          Switch  Interface
-----------------------------------------------------------------------
Te1/5/4   operational   0013.5fcb.1480  2    Te2/5/4
Te1/5/5   link_down     -               -     -

            Last operational         Current packet       Last Diag   Time since
Interface   Failure state            State                Result      Last Diag
-------------------------------------------------------------------------------
Te1/5/4 No failure               Hello bidir            Never ran   7M:51S
Te1/5/5 No failure               No failure             Never ran   7M:51S


                            Hello Tx (T4) ms        Hello Rx (T5*) ms
 Interface  State        Cfg     Cur     Rem     Cfg     Cur     Rem
----------------------------------------------------------------------
Te1/5/4 operational  500     500     404     5000    5000    4916
Te1/5/5 link_down    500     -       -       500000  -       -
Te2/5/4 operational  500     500     404     500000  500000  499916
Te2/5/5 link_down    500     -       -       500000  -       -
 *T5 = min_rx * multiplier
```

Configuring VSL QoS

The VSS automatically configures VSL ports for trust CoS, using default CoS mappings (you cannot change the mappings on VSL ports).

For switching modules that support per-ASIC configuration, the VSL configuration applies to all ports on the same ASIC (including any non-VSL ports).

The VSS disables the QoS commands on VSL ports (and any non-VSL ports on the same ASIC). For example, you cannot use QoS queuing or map commands on VSL ports.

To ensure that all eight QoS receive queues are enabled for the 10-Gigabit Ethernet ports on the supervisor engine, enter the **mls qos 10g-only** global configuration command.

In Cisco IOS Release 12.2(33)SXI and later releases, when the **mls qos 10g-only** command is entered and only one of the two 10-Gigabit Ethernet ports on the supervisor engine is a VSL port, the non-VSL 10-Gigabit Ethernet port can be configured for QoS.

Subcommands for VSL Port Channels

On a VSL port channel, only a subset of interface subcommands are available in the command console. Table 4-2 describes the available interface subcommands.

Table 4-2 Interface Subcommands for VSL Port Channels

Subcommand	Description
default	Sets a command to its defaults.
description	Enters a text description for the interface.
exit	Exits from interface configuration mode.
load-interval	Specifies interval for load calculation for an interface.
logging	Configures logging for interface.
mls	Specifies multilayer switching subcommands.
no	Disables a command, or sets the command defaults.
shutdown	Shuts down the selected interface.
switch virtual link	Specifies the switch associated with this port channel.
vslp	Specifies VSLP interface configuration commands.

Subcommands for VSL Ports

If a port is included in a VSL port channel, only a subset of interface subcommands are available in the command console. Table 4-3 describes the available interface subcommands.

Table 4-3 Interface Subcommands for VSL Ports

Subcommand	Description
channel-group	Adds the interface to the specified channel group.
default	Sets a command to its defaults.
description	Adds a description to the interface.
exit	Exits from interface configuration mode.
load-interval	Specifies interval for load calculation for an interface.
logging	Configures logging for the interface.
no	Disables a command, or sets the command defaults.
shutdown	Shuts down the selected interface.

Configuring the Router MAC Address Assignment

When the VSS is started for the first time, the initial VSS active supervisor engine assigns a router MAC address for the VSS. By default, the supervisor engine assigns a MAC address from its own chassis. After a switchover to the second chassis, the VSS continues to use the MAC address from the previously VSS active chassis as the router MAC address.

In the rare case where both chassis later become inactive, and then they start up with the second supervisor engine becoming the initial VSS active supervisor engine, the VSS will start up with a router MAC address from the second chassis. Other Layer 2 hosts that do not respond to GARP and are not directly connected to the VSS will retain the earlier router MAC address of the VSS, and will not be able to communicate with the VSS. To avoid this possibility, you can configure the VSS to assign a router MAC address from a reserved pool of addresses with the domain ID encoded in the last octet of the MAC address, or you can specify a MAC address.

Note If you change the router MAC address, you must reload the virtual switch for the new router MAC address to take effect.

To specify that the router MAC address is assigned from a reserved pool of domain-based addresses, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`switch virtual domain`** *`domain_id`*	Enters VSS configuration mode.
Step 2	`Router(config-vs-domain)#` **`mac-address use-virtual`**	The router MAC address is assigned from a reserved pool of domain-based addresses. **Note** The **no** form of this command reverts to the default setting, using a MAC address from the backplane of the initial VSS active chassis.

To specify a router MAC address, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`switch virtual domain`** *`domain_id`*	Enters VSS configuration mode.
Step 2	`Router(config-vs-domain)#` **`mac-address`** *`mac-address`*	The router MAC address is specified in three 2-byte hexadecimal numbers.

This example shows how to configure router MAC address assignment from a reserved pool of domain-based addresses:

```
Router(config)# switch virtual domain 255
Router(config-vs-domain)# mac-address use-virtual
```

The following example shows how to specify the router MAC address in hexadecimal format:

```
Router(config)# switch virtual domain 255
Router(config-vs-domain)# mac-address 0123.4567.89ab
```

Configuring Deferred Port Activation During VSS Standby Recovery

Instead of allowing all ports to be activated simultaneously when a failed chassis is restarted as the VSS standby chassis, you can configure the system to defer activation of non-VSL ports and then activate the ports in groups over a period of time.

To specify deferred port activation, perform this task:

	Command	Purpose
Step 1	Router(config)# **switch virtual domain 1**	Enters VSS configuration mode.
Step 2	Router(config-vs-domain)# **standby port delay** *delay-time*	Specifies that the port activation will be initially deferred and then performed in cycles. For *delay-time*, specify the period in seconds before port activation will begin. The range is 30 to 3600.
Step 3	Router(config-vs-domain)# **standby port bringup** *number cycle-time*	Specifies the number of ports to be activated per cycle and the waiting time between cycles. For *number*, specify the number of ports to be activated per cycle. The range is 1 to 100. The default value is 1 port. For *cycle-time*, specify the period in seconds between cycles. The range is 1 to 10. The default value is 1 second.

This example shows how to configure port activation to be deferred by 120 seconds, then activated in groups of 20 ports every 5 seconds:

```
Router(config)# switch virtual domain 1
Router(config-vs-domain)# standby port delay 120
Router(config-vs-domain)# standby port bringup 20 5
```

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 chassis are added to the EtherChannel. You can verify the MEC configuration by entering the **show etherchannel** command.

One VSS supports a maximum of 512 port channels.

Note Releases earlier than Cisco IOS Release 12.2(33)SXI support a maximum of 128 port channels.

The Configuring Port Load Share Deferral on the Peer Switch section provides additional details about MECs:

Configuring Port Load Share Deferral on the Peer Switch

To configure the load share deferral feature for a port channel, perform this task on the switch that is an MEC peer to the VSS:

	Command	Purpose
Step 1	`Router(config)# ` **`port-channel load-defer`** *`seconds`*	(Optional) Configures the port load share deferral interval for all port channels. • *seconds*—The time interval during which load sharing is initially 0 for deferred port channels. The range is 1 to 1800 seconds; the default is 120 seconds.
Step 2	`Router(config)# ` **`interface port-channel`** *`channel-num`*	Enters interface configuration mode for the port channel.
Step 3	`Router(config-if)# ` **`port-channel port load-defer`**	Enables port load share deferral on the port channel.

This example shows how to configure the load share deferral feature on port channel 10 of the switch that is an MEC peer to the VSS:

```
Router(config)# port-channel load-defer 60
Router(config)# interface port-channel 10
Router(config-if)# port-channel port load-defer
This will enable the load share deferral feature on this port-channel.
Do you wish to proceed? [yes/no]: yes
```

Configuring Dual-Active Detection

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

- Configuring Enhanced PAgP Dual-Active Detection, page 4-45
- Configuring BFD Dual-Active Detection, page 4-47
- Configuring Fast Hello Dual-Active Detection, page 4-48
- Configuring the Exclusion List, page 4-49
- Displaying Dual-Active Detection, page 4-49

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	`Router(config)# switch virtual domain domain_id`	Enters virtual switch submode.
Step 2	`Router(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	`Router(config)# switch virtual domain domain_id`	Enters virtual switch submode.
Step 2	`Router(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:

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

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

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

```
Router(config)# switch virtual domain 100
Router(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 BFD Dual-Active Detection

For the BFD dual-active detection, you must configure dual-active interface pairs that will act as BFD messaging links. By default, BFD detection is enabled.

To configure BFD dual-active detection, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`switch virtual domain`** *`domain_id`*	Enters virtual switch submode.
Step 2	`Router(config-vs-domain)#` **`dual-active detection bfd`**	Enables BFD dual-active detection method. By default, BFD detection is enabled.
Step 3	`Router(config-vs-domain)#` **`dual-active pair interface`** *`int_1`* **`interface`** *`int_2`* **`bfd`**	Configures the dual-active pair of interfaces. The interfaces *int_1* and *int_2* are of the form *`type`*[1] *`switch/slot/port`*. The interfaces must be directly connected (a single Layer 3 hop).

[1] *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

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

- The individual ports must be configured first with both an IP address and BFD configuration. The configuration is validated when you add the dual-active interface pair.
- The IP addresses assigned to the dual-active pair must be from two different networks or subnetworks.
- The BFD timers must be configured with the same values on the ports at both ends of the link to ensure proper operation of Layer 3 BFD dual-active detection.
- The MAC address cannot be specified on the interface.

Note We recommend that you configure a short BFD interval and small multiplier value (such as 50 to 100 ms for the interval and 3 as the multiplier value). If the interval and multiplier values are large, there is a long delay before the system initiates dual-active mode recovery. This condition can cause network instability and poor convergence.

This example shows how to configure interfaces for BFD dual-active detection:

```
Router (config)# interface gigabitethernet 1/9/48
Router (config-if)# no switchport
Router (config-if)# ip address 200.230.230.231 255.255.255.0
Router (config-if)# bfd interval 100 min_rx 100 multiplier 3
Router (config-if)# no shutdown
Router (config-if)# interface gigabitethernet 2/1/48
Router (config-if)# no switchport
Router (config-if)# ip address 201.230.230.231 255.255.255.0
Router (config-if)# bfd interval 100 min_rx 100 multiplier 3
Router (config-if)# no shutdown
Router (config-if)# exit
Router (config)# switch virtual domain 100
Router (config-vs-domain)# dual-active detection bfd
Router (config-vs-domain)# dual-active pair interface g 1/9/48 interface g 2/1/48 bfd

adding a static route 200.230.230.0 255.255.255.0 Gi2/1/48 for this dual-active pair
adding a static route 201.230.230.0 255.255.255.0 Gi1/9/48 for this dual-active pair
Router(config-vs-domain)# exit
Router(config)# exit
Router# show switch virtual dual-active bfd
Bfd dual-active detection enabled: Yes
Bfd dual-active interface pairs configured:
  interface1 Gi1/9/48 interface2 Gi2/1/48
```

Configuring Fast Hello Dual-Active Detection

Fast hello dual-active detection is enabled by default; however, you must configure dual-active interface pairs to act as fast hello dual-active messaging links.

To configure fast hello dual-active detection, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`switch virtual domain`** *`domain_id`*	Enters the virtual switch submode.
Step 2	`Router(config-vs-domain)#` **`dual-active detection fast-hello`**	Enables the fast hello dual-active detection method. Fast hello dual-active detection is enabled by default.
Step 3	`Router(config-vs-domain)#` **`exit`**	Exits virtual switch submode.
Step 4	`Router(config)#` **`interface`** *`type`*[1] *`switch/slot/port`*	Selects the interface to configure. This interface must be directly connected to the other chassis and must not be a VSL link.
Step 5	`Router(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	`Router(config-if)#` **`no shutdown`**	Activates the interface.

[1] *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

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.
- 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.
- Configuring fast hello dual-active mode will automatically remove all existing configuration from the interface and will restrict the interface to fast hello dual-active configuration commands.
- Unidirectional link detection (UDLD) will be disabled on fast hello dual-active interface pairs.

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

```
Router(config)# switch virtual domain 255
Router(config-vs-domain)# dual-active detection fast-hello
Router(config-vs-domain)# exit
Router(config)# interface fastethernet 1/2/40
Router(config-if)# dual-active fast-hello
```

```
WARNING: Interface FastEthernet1/2/40 placed in restricted config mode. All extraneous
configs removed!

Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# exit
Router# show run interface fastethernet 1/2/40
interface FastEthernet1/2/40
 no switchport
 no ip address
 dual-active fast-hello
end
```

Configuring the Exclusion List

When a dual-active scenario is detected, part of the recovery action is for the chassis to shut down all of its non-VSL interfaces. You can specify one or more interfaces to be excluded from this action (for example, to exclude the interface you use for remote access to the chassis).

To specify interfaces that are not to be shut down by dual-active recovery, perform this task:

	Command	Purpose
Step 1	Router(config)# **switch virtual domain** *domain_id*	Enters virtual switch submode.
Step 2	Router(config-vs-domain)# **dual-active exclude interface** *type*[1] *switch/slot/port*	Specifies an interface to exclude from shutting down in dual-active recovery.

[1] *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When you configure the exclusion list, note the following information:

- The interface must be a physical port configured with an IP address.
- The interface must not be a VSL port.
- The interface must not be in use for IP BFD dual-active detection.
- The interface must not be in use for fast hello dual-active detection.

This example shows how to configure an interface as an exclusion:

```
Router(config)# switch virtual domain 100
Router (config-vs-domain)# dual-active exclude interface gigabitethernet 1/5/5
```

Displaying Dual-Active Detection

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

Command	Purpose
Router# **show switch virtual dual-active** [**bfd** | **pagp** | **fast-hello** | **summary**]	Displays information about dual-active detection configuration and status.

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

```
Router# show switch virtual dual-active summary
Pagp dual-active detection enabled: Yes
Bfd dual-active detection enabled: Yes
```

```
Fast-hello dual-active detection enabled: Yes

No interfaces excluded from shutdown in recovery mode
In dual-active recovery mode: No
```

This example shows how to display information for BFD dual-active detection:

```
Router# show switch virtual dual-active bfd
Bfd dual-active detection enabled: Yes
Bfd dual-active interface pairs configured:
  interface1 Gi1/9/48 interface2 Gi2/1/48
```

This example shows how to display information for fast-hello dual-active detection:

```
Router# show switch virtual dual-active fast-hello
Fast-hello dual-active detection enabled: Yes

Fast-hello dual-active interfaces:
Port       State (local only)
-----------------------------
Gi1/4/47   Link dn
Gi2/4/47   -
```

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

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

Channel group 3 dual-active detect capability w/nbrs Dual-Active trusted group: No
          Dual-Active     Partner              Partner   Partner
Port      Detect Capable  Name                 Port      Version
Fa1/2/33  No              None                 None      N/A

Channel group 4
Dual-Active trusted group: Yes
No interfaces configured in the channel group

Channel group 5
Dual-Active trusted group: Yes
Channel group 5 is not participating in PAGP

Channel group 10 dual-active detect capability w/nbrs Dual-Active trusted group: Yes
          Dual-Active     Partner              Partner   Partner
Port      Detect Capable  Name                 Port      Version
Gi1/6/1   Yes             partner-1            Gi1/5/1   1.1
Gi2/5/1   Yes             partner-1            Gi1/5/2   1.1

Channel group 11 dual-active detect capability w/nbrs Dual-Active trusted group: No
          Dual-Active     Partner              Partner   Partner
Port      Detect Capable  Name                 Port      Version
Gi1/6/2   Yes             partner-1            Gi1/3/1   1.1
Gi2/5/2   Yes             partner-1            Gi1/3/2   1.1

Channel group 12 dual-active detect capability w/nbrs Dual-Active trusted group: Yes
          Dual-Active     Partner              Partner   Partner
Port      Detect Capable  Name                 Port      Version
Fa1/2/13  Yes             partner-1            Fa1/2/13  1.1
Fa1/2/14  Yes             partner-1            Fa1/2/14  1.1
Gi2/1/15  Yes             partner-1            Fa1/2/15  1.1
Gi2/1/16  Yes             partner-1            Fa1/2/16  1.1
```

Note The **show switch virtual dual-active pagp** command displays the same output as the **show pagp dual-active** command.

Configuring Service Modules in a VSS

To configure a service module in a VSS, you must add the switch number to many of the configuration commands, as described in this section.

Note For detailed instructions on configuring a service module in a VSS, see the configuration guide and command reference for the service module.

The following sections provide examples of how to configure a service module in a VSS:

- Opening a Session with a Service Module in a VSS, page 4-51
- Assigning a VLAN Group to a Firewall Service Module in a VSS, page 4-52
- Assigning a VLAN Group to an ACE Service Module in a VSS, page 4-52
- Verifying Injected Routes in a Service Module in a VSS, page 4-53

Opening a Session with a Service Module in a VSS

To configure service modules that require opening a session, perform this task:

Command	Purpose
Router# **session switch** *num* **slot** *slot* **processor** *processor-id*	Opens a session with the specified module. • *num*—Specifies the switch to access; valid values are 1 and 2. • *slot*—Specifies the slot number of the module. • *processor-id*—Specifies the processor ID number. Range: 0 to 9.

This example shows how to open a session to a Firewall Service Module in a VSS:

```
Router# session switch 1 slot 4 processor 1
The default escape character is Ctrl-^, then x.
You can also type 'exit' at the remote prompt to end the session
Trying 127.0.0.41 ... Open
```

Assigning a VLAN Group to a Firewall Service Module in a VSS

To assign a VLAN group to a FWSM, perform this task:

Command	Purpose
`Router(config)# firewall switch num slot slot vlan-group [vlan_group | vlan_range]`	Assigns VLANs to a firewall group in the specified module. • *num*—Specifies the switch to access; valid values are 1 and 2. • *slot*—Specifies the slot number of the module. • *vlan_group*—Specifies the group ID as an integer. • *vlan_range*—Specifies the VLANs assigned to the group.

This example shows how to assign a VLAN group to a Firewall Service Module in a VSS:

```
Router(config)# firewall switch 1 slot 4 vlan-group 100,200
```

Assigning a VLAN Group to an ACE Service Module in a VSS

To assign a VLAN group to an ACE, perform this task:

	Command	Purpose
Step 1	`Router(config)# svclc multiple-vlan-interfaces`	Enables multiple VLAN interfaces mode for service modules.
Step 2	`Router(config)# svclc switch num slot slot vlan-group [vlan_group | vlan_range]`	Assign VLANs to a firewall group in the specified module. • *num*—Specifies the switch to access; valid values are 1 and 2. • *slot*—Specifies the slot number of the module. • *vlan_group*—Specifies the group ID as an integer. • *vlan_range*—Specifies the VLANs assigned to the group.

This example shows how to assign multiple VLAN groups to an ACE service module in a VSS:

```
Router(config)# svclc multiple-vlan-interfaces
Router(config)# svclc switch 1 slot 4 vlan-group 100,200
```

Verifying Injected Routes in a Service Module in a VSS

To view route health injection (RHI) routes, perform this task:

Command	Purpose
`Router# show svclc rhi-routes switch num slot slot`	Displays injected RHI routes in the specified service module. • *num*—Specifies the switch to access; valid values are 1 and 2. • *slot*—Specifies the slot number of the module.

This example shows how to view injected routes in a service module in a VSS:

```
Router# show svclc rhi-routes switch 1 slot 4
RHI routes added by slot 34
        ip             mask           nexthop      vlan  weight tableid
  --------------- --------------- --------------- ------ ------ -------
A 23.1.1.4        255.255.255.252 20.1.1.1        20     1      0
```

Viewing Chassis Status and Module Information in a VSS

To view chassis status and information about modules installed in either or both chassis of a VSS, perform the following task:

Command	Purpose
`Router# show module switch { 1 | 2 | all }`	Displays information about modules in the specified chassis (**1** or **2**), or in both chassis (**all**).

This example shows how to view the chassis status and module information for chassis number 1 of a VSS:

```
Router# show module switch 1
 Switch Number:     1   Role:   Virtual Switch Active
----------------------  -----------------------------
Mod Ports Card Type                              Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------
  1   48  CEF720 48 port 10/100/1000mb Ethernet  WS-X6748-GE-TX     SAL1215M2YA
  2    8  CEF720 8 port 10GE with DFC            WS-X6708-10GE      SAL1215M55F
  3    1  Application Control Engine Module      ACE20-MOD-K9       SAD120603SU
.
.
.
```

Upgrading a VSS

Cisco IOS Rel ease 12.2(33)SXH supports a fast software upgrade (FSU) of the VSS using RPR. Cisco IOS Release 12.2(33)SXI and later releases support an enhanced fast software upgrade (eFSU) of the VSS using SSO.

This section describes both types of VSS upgrade:

- Performing a Fast Software Upgrade of a VSS, page 4-54
- Performing an Enhanced Fast Software Upgrade of a VSS, page 4-55

Performing a Fast Software Upgrade of a VSS

The FSU of a VSS is similar to the RPR-based standalone chassis FSU described in the "Performing a Fast Software Upgrade" section on page 7-6. While the standalone chassis upgrade is initiated by reloading the VSS standby supervisor engine, the VSS upgrade is initiated by reloading the VSS standby chassis. During the FSU procedure, a software version mismatch between the VSS active and the VSS standby chassis causes the system to boot in RPR redundancy mode, which is stateless and causes a hard reset of the all modules. As a result, the FSU procedure requires system downtime corresponding to the RPR switchover time.

Note VSS mode supports only one supervisor engine in each chassis. If another supervisor engine resides in the chassis it will act as the DFC.

To perform an FSU of a VSS, perform this task:

	Command	Purpose
Step 1	`Router# copy tftp disk_name`	Uses TFTP to copy the new software image to flash memory on the VSS active and VSS standby chassis (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# no boot system`	Removes any previously assigned boot variables.
Step 4	`Router(config)# config-register 0x2102`	Sets the configuration register.
Step 5	`Router(config)# boot system flash device:file_name`	Configures the chassis to boot the new image.
Step 6	`Router(config)# end`	Returns to privileged EXEC mode.
Step 7	`Router# copy running-config startup-config`	Saves the configuration.

	Command	Purpose
Step 8	`Router# redundancy reload peer`	Reloads the VSS standby chassis and brings it back online running the new version of the Cisco IOS software. Due to the software version mismatch between the two chassis, the VSS standby chassis will be in RPR redundancy mode. **Note** Before reloading the VSS standby chassis, make sure you wait long enough to ensure that all configuration synchronization changes have completed.
Step 9	`Router# redundancy force-switchover`	Forces the VSS standby chassis to assume the role of the VSS active chassis running the new Cisco IOS image. The modules are reloaded and the module software is downloaded from the new VSS active chassis. The old VSS active chassis reboots with the new image and becomes the VSS standby chassis.

This example shows how to perform an FSU:

```
Router# config terminal
Router(config)# no boot system
Router(config)# config-register 0x2102
Router(config)# boot system flash disk0:image_name
Router(config)# end
Router# copy running-config startup-config
Router# redundancy reload peer
Router# redundancy force-switchover
```

Performing an Enhanced Fast Software Upgrade of a VSS

An eFSU uses the same commands and software infrastructure as an in-service software upgrade (ISSU). The eFSU differs from an ISSU in that it resets the modules, which results in a brief traffic interruption. The eFSU sequence for a VSS follows the same logical steps as the single-chassis eFSU described in the "Performing an Enhanced Fast Software Upgrade" section on page 5-5, but the procedure applies to the VSS active and VSS standby supervisor engine in each chassis, instead of two supervisor engines in one chassis. During an eFSU, the VSS standby chassis, including the supervisor engine and modules, is upgraded and brought up in a stateful switchover (SSO) mode. The eFSU process then forces a switchover and performs the same upgrade on the other chassis, which becomes the new VSS standby.

Note VSS mode supports only one supervisor engine in each chassis. If another supervisor resides in the chassis it will act as the DFC.

This section contains the following topics:

- eFSU Restrictions and Guidelines, page 4-56
- eFSU Stages for a VSS Upgrade, page 4-57
- Configuring and Performing an eFSU Upgrade, page 4-58
- eFSU Upgrade Example, page 4-66

eFSU Restrictions and Guidelines

When performing an eFSU, note the following guidelines and restrictions:

- 7600-SIP-400 is powered down during an eFSU and is powered up at the Commitversion Stage or at Abortversion (Optional).
- An eFSU can install a full image upgrade or a patch upgrade. Any patch upgrade will be installed by the same process as a full image upgrade, regardless of whether the patch requires a reload or a process restart.
- The new image file must reside in the file system of the supervisor engine in each chassis before the eFSU can be initiated. The **issu** commands will accept only global file system names (for example, disk0: or sup-bootdisk:). The **issu** commands will not accept switch number-specific file system names (for example, sw1-slot5-disk0:).
- When preparing for the eFSU, do not change the boot variable. Although a boot variable change is required in the FSU (RPR) procedure, changing the boot variable in the eFSU procedure will cause the CurrentVersion variable to be inconsistent, preventing execution of the eFSU.
- The **issu** commands for a VSS eFSU upgrade are similar to those for a single-chassis (standalone) eFSU, as described in the "Performing an Enhanced Fast Software Upgrade" section on page 5-5, with the following differences:
  - Where the standalone **issu** commands accept an argument of slot number, the VSS **issu** commands accept a switch and slot number, in the format of *switch/slot* (for example, 1/5 refers to switch 1, slot 5).
  - For a normal VSS eFSU, it is not necessary to specify a switch or slot number when entering the VSS **issu** commands.
- You cannot change the rollback timer period during the eFSU process.
- During the eFSU process, do not perform any manual switchover other than those caused by the **issu** commands.
- During the eFSU process, do not perform an online insertion or removal (OIR) of any module.
- During an eFSU downgrade, if the process is aborted (either due to an MCL error or by entering the **abortversion** command) just after executing the **loadversion** command, the SSO VSS standby is reloaded with the original image but the SSO VSS standby's ICS is not because the bootvar of the SSO VSS standby's ICS is not modified during an abort executed after the **loadversion** command.
- Images with different feature sets fail the eFSU compatibility check, regardless of the software release.
- The eFSU feature does not support upgrades or downgrades between modular and non-modular IOS versions.
- The eFSU feature does not support upgrades or downgrades between installed and binary modes of modular IOS. The Installed mode was removed after Cisco IOS Release 12.2(33)SXI3.
- Before you start a downgrade with eFSU (reverting to an earlier version of Cisco IOS software), remove configurations and disable any features or functions that are not supported in the earlier version. Otherwise the configuration files fail to synchronize and the standby supervisor engine reloads.
- The eFSU upgrade feature works with NSF/SSO. Software features that do not support NSF/SSO stop operating until after the software upgrade switchover, when they come back online.
- Images with release dates more than 18 months apart are not supported for eFSU. See the SX_SY_EFSU_Compatibility_Matrix to verify compatibility.

eFSU Stages for a VSS Upgrade

The eFSU sequence consists of several stages, each explicitly initiated by entering a specific **issu** command in the CLI. At each stage, you can verify the system status or roll back the upgrade before moving to the next stage.

The following sections describe the eFSU stages for a VSS upgrade:

- Preparation, page 4-57
- Loadversion Stage, page 4-57
- Runversion Stage, page 4-57
- Acceptversion Stage (Optional), page 4-57
- Commitversion Stage, page 4-58
- Abortversion (Optional), page 4-58

Preparation

Before you can initiate the eFSU process, the upgrade image must reside in the file system of the supervisor engine in each chassis; otherwise, the initial command will be rejected. The VSS must be in a stable operating state with one chassis in the VSS active state and the other chassis in the hot VSS standby state.

Loadversion Stage

The eFSU process begins when you enter the **issu loadversion** command specifying the location in memory of the new upgrade images on the VSS active and VSS standby chassis. Although the **issu loadversion** command allows you to specify the switch and slot number of the VSS active and VSS standby chassis, it is not necessary to do so. When you enter the **issu loadversion** command, the entire VSS standby chassis, including the supervisor engine and modules, is reloaded with the new upgrade image. Because the VSS standby chassis modules are unavailable while reloading, the throughput of the VSS is temporarily reduced by 50 percent during this stage. After reloading, the VSS standby chassis boots with the new image and initializes in SSO mode, restoring traffic throughput. In this state, the VSS standby chassis runs a different software version than the VSS active chassis, which requires the VSS active chassis to communicate with modules running different image versions between the two chassis.

Runversion Stage

When the VSS standby chassis is successfully running the new image in SSO mode, you can enter the **issu runversion** command. This command forces a switchover in which the upgraded VSS standby chassis takes over as the new VSS active chassis. The formerly VSS active chassis reloads and initializes as the new VSS standby chassis in SSO mode, running the old image. As in the loadversion stage, the throughput of the VSS is temporarily reduced during the VSS standby chassis reload, and the VSS standby chassis runs a different software version than the VSS active chassis.

Acceptversion Stage (Optional)

When you enter the **issu runversion** command, a switchover to the chassis running the new image occurs, which starts an automatic rollback timer as a safeguard to ensure that the upgrade process does not cause the VSS to be nonoperational. Before the rollback timer expires, you must either accept or commit the new software image. If the timer expires, the upgraded chassis reloads and reverts to the previous software version. To stop the rollback timer, enter the **issu acceptversion** command. Prior to starting the eFSU process, you can disable the rollback timer or configure it to a value up to two hours (the default is 45 minutes).

Operating with an upgraded VSS active chassis, this stage allows you to examine the functionality of the new software image. When you are satisfied that the new image is acceptable, enter the **issu commitversion** command to complete the upgrade process.

Commitversion Stage

To apply the upgrade image to the second chassis, completing the eFSU, enter the **issu commitversion** command. The VSS standby chassis is reloaded and booted with the new upgrade image, initializing again as the VSS standby chassis. As in the loadversion stage, the throughput of the VSS is temporarily reduced while the modules are reloaded and initialized. After the successful reload and reboot of the VSS standby chassis, the VSS upgrade process is complete.

Abortversion (Optional)

At any time before you enter the **issu commitversion** command, you can roll back the upgrade by entering the **issu abortversion** command. The upgrade process also aborts automatically if the software detects a failure. The rollback process depends on the current state. If the eFSU is aborted before you enter the **issu runversion** command, the VSS standby chassis is reloaded with the old image. If the eFSU is aborted after the **issu runversion** command, a switchover is executed. The VSS standby chassis, running the old image, becomes the VSS active chassis. The formerly VSS active chassis is then reloaded with the old image, completing the rollback.

Configuring and Performing an eFSU Upgrade

The following sections describe how to configure and perform an eFSU upgrade:

- Changing the eFSU Rollback Timer, page 4-59
- Performing an eFSU Upgrade, page 4-59
- Performing an eFSU Upgrade from Previous Cisco IOS Releases to Cisco IOS Release 12.2(33)SXI4, page 4-60
- Performing an eFSU Upgrade from Cisco IOS Release 12.2(33)SXI4 to Future Cisco IOS Releases, page 4-61
- Performing an eFSU Downgrade from Cisco IOS Release 12.2(33)SXI4 to Earlier Cisco IOS Releases, page 4-62
- Performing an eFSU Downgrade from a Future Cisco IOS Release to Cisco IOS Release 12.2(33)SXI4, page 4-64
- Performing an eFSU Upgrade on an Installed Modular Image, page 4-65
- Aborting an eFSU Upgrade, page 4-66

Changing the eFSU Rollback Timer

To view or change the eFSU rollback timer, perform the following task before beginning an upgrade:

	Command	Purpose
Step 1	`Router#` **`config terminal`**	Enters configuration mode.
Step 2	`Router(config)#` **`issu set rollback-timer`** `{`*`seconds`* `|` *`hh:mm:ss`*`}`	(Optional) Sets the rollback timer to ensure that the upgrade process does not leave the VSS nonoperational. If the timer expires, the software image reverts to the previous software image. To stop the timer, you must either accept or commit the new software image. The timer duration can be set with one number (*seconds*), indicating the number of seconds, or as hours, minutes, and seconds with a colon as the delimiter (*hh:mm:ss*). The range is 0 to 7200 seconds (2 hours); the default is 2700 seconds (45 minutes). A setting of 0 disables the rollback timer.
Step 3	`Router(config)#` **`exit`**	Returns to privileged EXEC mode.
Step 4	`Router#` **`show issu rollback timer`**	Displays the current rollback timer value.

This example shows how to set the eFSU rollback timer to one hour using both command formats:

```
Router# config terminal
Router(config)# issu set rollback-timer 3600
% Rollback timer value set to [ 3600 ] seconds
Router(config)# issu set rollback-timer 01:00:00
% Rollback timer value set to [ 3600 ] seconds
Router(config)#
```

Performing an eFSU Upgrade

To perform an eFSU upgrade (or downgrade) of a VSS, perform this task:

	Command	Purpose
Step 1	`Router#` **`copy tftp`** *`disk_name`*	Uses TFTP to copy the new software image to flash memory on the VSS active and VSS standby chassis (disk0: and slavedisk0:) and to the ICS's, if they exist. Answer the prompts to identify the name and location of the new software image.
Step 2	`Router#` **`show issu state`** `[`*`switch/slot`*`] [`**`detail`**`]`	(Optional) Verifies that the VSS is ready to run the eFSU. **Note** You can use the **show issu state** command at any stage of the upgrade to verify the progress and status of the upgrade.
Step 3	`Router#` **`issu loadversion`** `[`*`active_switch/slot`*`]` *`active-image`* `[`*`standby_switch/slot`*`]` *`standby-image`*	Starts the upgrade process by loading the new software image onto the VSS standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the VSS standby chassis to transition to SSO mode.

	Command	Purpose
Step 4	`Router# `**`issu runversion`**	Forces a switchover, causing the VSS standby chassis to become VSS active and begin running the new software. The previously VSS active chassis becomes VSS standby and boots with the old image.
Step 5	`Router# `**`issu acceptversion`**	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.
Step 6	`Router# `**`issu commitversion`**	Loads the new software image onto the VSS standby chassis.
Step 7	`Router# `**`show issu state`**` [`*`switch/slot`*`][`**`detail`**`]`	Verifies the status of the upgrade process. If the upgrade was successful, both the VSS active and VSS standby chassis are running the new software version.

For an example of the eFSU upgrade sequence, see the "eFSU Upgrade Example" section on page 4-66.

Performing an eFSU Upgrade from Previous Cisco IOS Releases to Cisco IOS Release 12.2(33)SXI4

With previous Cisco IOS releases if you have a second ICS in your chassis, it will be forced to ROMMON.

To perform an eFSU upgrade of a VSS from Cisco IOS Release 12.2(33)SXI to Cisco IOS Release 12.(33)SXI4, perform this task:

	Command	Purpose
Step 1	`Router# `**`copy tftp`**` `*`disk_name`*	Uses TFTP to copy the new software image to flash memory on the active and standby chassis (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image.
Step 2	`Router# `**`show issu state`**` [`*`switch/slot`*`] [`**`detail`**`]`	(Optional) Verifies that the VSS is ready to run the eFSU. **Note** You can use the **show issu state** command at any stage of the upgrade to verify the progress and status of the upgrade.

Step 3	`Router# `**`issu loadversion`**
	`[`*`active_switch/slot`*`] `*`active-image`*
	`[`*`standby_switch/slot`*`] `*`standby-image`*
		Starts the upgrade process by loading the new software image onto the standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*.
		It may take several seconds for the new image to load and for the standby chassis to transition to SSO mode.

Step 4	`Router# `**`issu runversion`**	Forces a switchover, causing the standby chassis to become active and begin running the new software. The previously active chassis becomes standby and boots with the old image.
Step 5	`Router# `**`issu acceptversion`**	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.

	Command	Purpose
Step 6	`Router#` **`issu commitversion`**	Loads the new software image onto the standby chassis.
Step 7	`Router#` **`show issu state`** [*`switch/slot`*][**`detail`**]	Verifies the status of the upgrade process. If the upgrade was successful, both the active and standby chassis are running the new software version.

If you intend to bring up the ICS supervisor engine with Cisco IOS Release 12.2(33)SXI4, you will need to manually boot up the ICS supervisor engine after the eFSU cycle is complete.

Performing an eFSU Upgrade from Cisco IOS Release 12.2(33)SXI4 to Future Cisco IOS Releases

To perform an eFSU upgrade of a VSS from Cisco IOS Release 12.2(33)SXI4 to a future Cisco IOS Release, perform this task:

	Command	Purpose
Step 1	`Router#` **`copy tftp`** *`disk_name`*	Uses TFTP to copy the new software image to flash memory on the active and standby chassis (disk0: and slavedisk0:) and to the ICSs, if they exist. Answer the prompts to identify the name and location of the new software image.
Step 2	`Router#` **`show issu state`** [*`switch/slot`*] [**`detail`**]	(Optional) Verifies that the VSS is ready to run the eFSU. **Note** You can use the **show issu state** command at any stage of the upgrade to verify the progress and status of the upgrade.
Step 3	`Router#` **`switch virtual in-chassis standby`** *`switch`* [**`disable`** | **`enable`**]	(Optional) Includes or removes the ICS, from the eFSU cycle. This command must be executed before the start of the eFSU cycle even if the ICS is in ROMMON.
Step 4	`Router#` **`issu loadversion`** [*`active_switch/slot`*] *`active-image`* [*`standby_switch/slot`*] *`standby-image`*	Starts the upgrade process by loading the new software image onto the standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the standby chassis to transition to SSO mode. **Note** This command is not extended for the ICS. The file system mentioned for the ICA is used for the respective ICS. For example, if the **issu loadversion disk0:***image_name* **slavesup-bootdisk:***image_name* command is executed the loadversion command is accepted. The presence of the image is checked in the disk0: for the active supervisor engine (both the ICA and ICS) and the SPs bootdisk for the SSO standby (both the ICA and ICS).

	Command	Purpose
Step 5	Router# **issu runversion**	Forces a switchover, causing the standby engine chassis to become active and begin running the new software. The previously active chassis becomes standby and boots with the old image. **Note** If there are two supervisor engines in the active chassis, an in-chassis role reversal will occur if the upgrade cycle starts with both supervisor engines in the active chassis unless you have configured a supervisor engine to not participate in the upgrade.
Step 6	Router# **issu acceptversion**	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.
Step 7	Router# **issu commitversion**	Loads the new software image onto the standby chassis.
Step 8	Router# **show issu state** [*switch/slot*][**detail**]	Verifies the status of the upgrade process. If the upgrade was successful, both the active and standby chassis are running the new software version. The ICS is forced to ROMMON.

If the ICS is participating in the eFSU upgrade, you must ensure that the ICS is up and running before performing each ISSU step. If the ICS is not up and running, you need to wait until it is online. You can verify that the ICS is online by entering the **show module** command.

Performing an eFSU Downgrade from Cisco IOS Release 12.2(33)SXI4 to Earlier Cisco IOS Releases

To perform an eFSU downgrade of a VSS from Cisco IOS Release 12.2(33)SXI4 to an earlier Cisco IOS release, perform this task:

	Command	Purpose
Step 1	Router# **copy tftp** *disk_name*	Uses TFTP to copy the new software image to flash memory on the active and standby chassis (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image.
Step 2	Router# **show issu state** [*switch/slot*] [**detail**]	(Optional) Verifies that the VSS is ready to run the eFSU. **Note** You can use the **show issu state** command at any stage of the upgrade to verify the progress and status of the upgrade.
Step 3	Router# **switch virtual in-chassis standby** *switch* [**disable** | **enable**]	(Optional) Includes or removes the ICS from the eFSU cycle. This command must be executed before the start of the eFSU cycle even if the ICS is in ROMMON.

	Command	Purpose
Step 4	`Router# issu loadversion [active_switch/slot] active-image [standby_switch/slot] standby-image`	Starts the downgrade process by loading the new software image onto the standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the standby chassis to transition to SSO mode. **Note** If the active ICS is online when you enter the **issu loadversion** command, then an error message is displayed when the standby supervisor engine is booting up with the pre-12.2(33)SXI4 image, which prompts you to disable the active ICS. Once you disable the active ICS, the cycle will proceed. If you do not disable the active ICS and enter the **issu runversion** command, the command is not accepted. You will have to either abort the downgrade process or disable the active ICS to proceed with the downgrade. If the standby ICS is online when you enter the **issu loadversion** command, the pre-12.2(33)SXI4 image that comes up on the SSO standby forces the standby ICS to ROMMON.
Step 5	`Router# issu runversion`	Forces a switchover, causing the standby chassis to become active and begin running the new software. The previously active chassis becomes standby and boots with the old image.
Step 6	`Router# issu acceptversion`	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.
Step 7	`Router# issu commitversion`	Loads the new software image onto the standby chassis.
Step 8	`Router# show issu state [switch/slot][detail]`	Verifies the status of the downgrade process. If the downgrade was successful, both the active and standby chassis are running the new software version.

Performing an eFSU Downgrade from a Future Cisco IOS Release to Cisco IOS Release 12.2(33)SXI4

To perform an eFSU downgrade of a VSS from a future Cisco IOS Release to Cisco IOS Release 12.2(33)SXI4, perform this task:

	Command	Purpose
Step 1	Router# **copy tftp** *disk_name*	Uses TFTP to copy the new software image to the ICSs and flash memory on the active and standby chassis (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image.
Step 2	Router# **show issu state** [*switch/slot*] [**detail**]	(Optional) Verifies that the VSS is ready to run the eFSU. **Note** You can use the **show issu state** command at any stage of the upgrade to verify the progress and status of the upgrade.
Step 3	Router# **switch virtual in-chassis standby** *switch* [**disable** | **enable**]	(Optional) Includes or removes the ICS from the eFSU cycle. This command must be executed before the start of the eFSU cycle even if the ICS is in ROMMON. **Note** If you did not remove the ICS from the downgrade using the switch virtual disable command the loadversion cycle is aborted, the SSO standby reloads with the initial image.
Step 4	Router# **issu loadversion** [*active_switch/slot*] *active-image* [*standby_switch/slot*] *standby-image*	Starts the downgrade process by loading the new software image onto the standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the standby chassis to transition to SSO mode. **Note** This command is not extended for the ICS. The file system mentioned for the ICA is used for the respective ICS. For example, if the **issu loadversion disk0:***image_name* **slavesup-bootdisk:***image_name* command is executed the loadversion command is accepted. The presence of the image is checked in the disk0: for the active supervisor engine (both the ICA and ICS) and the SPs bootdisk for the SSO standby (both the ICA and ICS).
Step 5	Router# **issu runversion**	Forces a switchover, causing the standby chassis to become active and begin running the new software. The previously active chassis becomes standby and boots with the old image.
Step 6	Router# **issu acceptversion**	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.

	Command	Purpose
Step 7	Router# **issu commitversion**	Loads the new software image onto the standby chassis.
Step 8	Router# **show issu state** [*switch/slot*][**detail**]	Verifies the status of the downgrade process. If the downgrade was successful, both the active and standby chassis are running the new software version.

If the ICS is participating in the eFSU upgrade, you must ensure that the ICS is up and running before performing each ISSU step. If the ICS is not up and running you need to wait until it is online. You can verify that the ICS is online by entering the **show module** command.

Performing an eFSU Upgrade on an Installed Modular Image

To perform an eFSU upgrade (or downgrade) of an ION VSS, perform this task:

	Command	Purpose
Step 1	Router# **copy tftp** *disk_name*	Uses TFTP to copy the new software image to flash memory on the active and VSS standby chassis (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image. **Note** You should have a console on both the active and VSS standby supervisor engines because you will go back and forth between them.
Step 2	Router# **install file bootdisk:***filename* **bootdisk:***/location*	Installs the modular image on to both the active and VSS standby supervisor engines.
Step 3	Router# **show issu state** [*switch/slot*][**detail**]	Verifies the status of the upgrade process; status should display ‘Init’.
Step 4	Router# **issu loadversion** *new-image*	Starts the upgrade process by loading the installed software image onto the active and VSS standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the VSS standby chassis to transition to SSO mode. **Note** This command will cause the VSS standby chassis to reload.
Step 5	Router# **show issu state** [*switch/slot*][**detail**]	Verifies the status of the upgrade process; status should display ‘Load Version’.
Step 6	Router# **issu runversion**	Forces a switchover, causing the VSS standby chassis to become active and begin running the new software. The previously active chassis becomes VSS standby and boots with the old image.
Step 7	Router# **show issu state** [*switch/slot*][**detail**]	Verifies the status of the upgrade process; status should display ‘Run Version’.

	Command	Purpose
Step 8	`Router# issu commitversion`	Loads the new software image onto the VSS standby chassis. **Note** This command will cause the VSS standby chassis to reload.
Step 9	`Router# show issu state [switch/slot][detail]`	Verifies the status of the upgrade process; status should display 'Init'.
Step 10	`Router# redundancy force-switchover`	(Optional) Forces the VSS standby Route Processor (RP) to assume the role of the active RP.

For an example of the eFSU upgrade on an Installed Modular Image sequence, see the "eFSU Upgrade on an Installed Modular Image Example" section on page 4-67.

Aborting an eFSU Upgrade

To manually abort the eFSU and roll back the upgrade, perform this task:

Command	Purpose
`Router# issu abortversion`	Stops the upgrade process and forces a rollback to the previous software image.

This example shows how to abort an eFSU upgrade for a VSS:

```
Router# issu abortversion
```

eFSU Upgrade Example

This example shows how to perform and verify an eFSU upgrade for a VSS.

Verify the System Readiness

After copying the new image files into the file systems of the active and VSS standby chassis, enter the **show issu state detail** command and the **show redundancy status** command to check that the VSS is ready to perform the eFSU. One chassis must be in the active state and the other chassis in the hot VSS standby state. Both chassis should be in the ISSU Init state and in SSO redundancy state. In the example, both chassis are running an "oldversion" image.

```
Router# show issu state detail
                          Slot = 1/2
                      RP State = Active
                    ISSU State = Init
                 Boot Variable = disk0:s72033-oldversion.v1,12;
                Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:s72033-oldversion.v1
                Variable Store = PrstVbl

                          Slot = 2/7
                      RP State = Standby
                    ISSU State = Init
                 Boot Variable = disk0:s72033-oldversion.v1,12;
                Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:s72033-oldversion.v1

Router# show redundancy status
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Secondary
        Unit ID = 18

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
 Communications = Up

   client count = 132
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
    keep_alive threshold = 18
           RF debug mask = 0x0
```

eFSU Upgrade on an Installed Modular Image Example

This example shows how to perform an an eFSU upgrade on an installed modular image:

```
Router# copy ftp://172.18.108.26/s72033-advipservicesk9_wan-vz.122-33.SXI2
sup-bootdisk:s72033-advipservicesk9_wan-vz.122-33.SXI2.bin
Router# copy ftp://172.18.108.26/s72033-advipservicesk9_wan-vz.122-33.SXI2
slavesup-bootdisk:s72033-advipservicesk9_wan-vz.122-33.SXI2.bin
Router# install file sup-bootdisk:s72033-advipservicesk9_wan-vz.122-33.SXI2.bin sup-bootdisk:/newsys
Router# install file slavesup-bootdisk:s72033-advipservicesk9_wan-vz.122-33.SXI2.bin
slavesup-bootdisk:/newsys
Router# show issu state
                      Slot = 1/6
                  RP State = Active
                ISSU State = Init
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;

                      Slot = 2/6
                  RP State = Standby
                ISSU State = Init
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;
Router# issu loadversion sup-bootdisk:/newsys/s72033/base/s72033-advipservicesk9_wan-vm
%issu loadversion executed successfully, Standby is being reloaded
Router# show issu state
                      Slot = 1/6
                  RP State = Active
                ISSU State = Load Version
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;

                      Slot = 2/6
                  RP State = Standby
                ISSU State = Load Version
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;
Router# issu runversion
This command will reload the Active unit. Proceed ? [confirm]
Router# show issu state
                      Slot = 2/6
                  RP State = Active
                ISSU State = Run Version
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;

                      Slot = 1/6
                  RP State = Standby
                ISSU State = Run Version
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;
Router# issu commitversion
%issu commitversion executed successfully
Router# show issu state
                      Slot = 2/6
                  RP State = Active
                ISSU State = Init
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;

                      Slot = 1/6
                  RP State = Standby
                ISSU State = Init
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;
Router# redundancy force-switchover
```

Load the New Image to the VSS Standby Chassis

Enter the **issu loadversion** command to start the upgrade process. In this step, the VSS standby chassis reboots, reloads with the new image, and initializes as the VSS standby chassis in SSO redundancy mode, running the new image. This step is complete when the chassis configuration is synchronized, as indicated by the "Bulk sync succeeded" message.

```
Router# issu loadversion disk0:s72033-newversion.v2

000133: Aug  6 16:17:44.486 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/2/4, changed state to down
000134: Aug  6 16:17:43.507 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet2/7/4, changed state to down
000135: Aug  6 16:17:43.563 PST: %LINK-3-UPDOWN: Interface TenGigabitEthernet2/7/4,
changed state to down
000136: Aug  6 16:17:44.919 PST: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/2/4,
changed state to down
```

(Deleted many interface and protocol down messages)

```
%issu loadversion executed successfully, Standby is being reloaded
```

(Deleted many interface and protocol down messages, then interface and protocol up messages)

```
0000148: Aug  6 16:27:54.154 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/2/5, changed state to up
000149: Aug  6 16:27:54.174 PST: %LINK-3-UPDOWN: Interface TenGigabitEthernet2/7/5,
changed state to up
000150: Aug  6 16:27:54.186 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet2/7/5, changed state to up
000151: Aug  6 16:32:58.030 PST: %HA_CONFIG_SYNC-6-BULK_CFGSYNC_SUCCEED: Bulk Sync
succeeded
```

Verify the New Image on the VSS Standby Chassis

You can now enter the **show issu state detail** command and the **show redundancy** command to check that both chassis are in the ISSU Load Version state and SSO redundancy state. In this example, the VSS standby chassis is now running the “newversion” image.

```
Router# show issu state detail
                          Slot = 1/2
                      RP State = Active
                    ISSU State = Load Version
                 Boot Variable = disk0:s72033-oldversion.v1,12
                Operating Mode = sso
               Primary Version = disk0:s72033-oldversion.v1
             Secondary Version = disk0:s72033-newversion.v2
               Current Version = disk0:s72033-oldversion.v1
                Variable Store = PrstVbl

                          Slot = 2/7
                      RP State = Standby
                    ISSU State = Load Version
                 Boot Variable =
disk0:s72033-newversion.v2,12;disk0:s72033-oldversion.v1,12
                Operating Mode = sso
               Primary Version = disk0:s72033-oldversion.v1
             Secondary Version = disk0:s72033-newversion.v2
               Current Version = disk0:s72033-newversion.v2

Router# show redundancy status
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Secondary
        Unit ID = 18

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
 Communications = Up

   client count = 132
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 18
           RF debug mask = 0x0
```

Execute a Switchover to the New Image

When the VSS standby chassis is successfully running the new image in SSO redundancy state, enter the **issu runversion** command to force a switchover. The upgraded VSS standby chassis takes over as the new active chassis, running the new image. The formerly active chassis reloads and initializes as the new VSS standby chassis in SSO mode, running the old image (in case the software upgrade needs to be aborted and the old image restored). This step is complete when the chassis configuration is synchronized, as indicated by the “Bulk sync succeeded” message.

```
Router# issu runversion
This command will reload the Active unit.  Proceed ? [confirm]
System Bootstrap, Version 12.2(17r)S4, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2005 by cisco Systems, Inc.
```

```
Cat6k-Sup720/RP platform with 1048576 Kbytes of main memory

Download Start
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

(Deleted many lines)

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Download Completed! Booting the image.
Self decompressing the image :
##########################################################################################
```

(Deleted many lines)

```
################################################################################ [OK]
running startup....
```

(Deleted many lines)

```
000147: Aug  6 16:53:43.199 PST: %HA_CONFIG_SYNC-6-BULK_CFGSYNC_SUCCEED: Bulk Sync
succeeded
```

Verify the Switchover

You can now enter the **show issu state detail** command and the **show redundancy** command to check that both chassis are in the ISSU Run Version state and SSO redundancy state. In this example, the active chassis is now running the “newversion” image.

```
Router# show issu state detail
                          Slot = 2/7
                      RP State = Active
                    ISSU State = Run Version
                 Boot Variable =
disk0:s72033-newversion.v2,12;disk0:s72033-oldversion.v1,12
                Operating Mode = sso
               Primary Version = disk0:s72033-newversion.v2
             Secondary Version = disk0:s72033-oldversion.v1
               Current Version = disk0:s72033-newversion.v2
                Variable Store = PrstVbl

                          Slot = 1/2
                      RP State = Standby
                    ISSU State = Run Version
                 Boot Variable = disk0:s72033-oldversion.v1,12
                Operating Mode = sso
               Primary Version = disk0:s72033-newversion.v2
             Secondary Version = disk0:s72033-oldversion.v1
               Current Version = disk0:s72033-oldversion.v1

Router# show redundancy status
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Primary
        Unit ID = 39

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
 Communications = Up

   client count = 134
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 18
           RF debug mask = 0x0
```

Commit the New Image to the VSS Standby Chassis

When the active chassis is successfully running the new image in the SSO redundancy state, you can enter either the **issu acceptversion** command to stop the rollback timer and hold this state indefinitely, or the **issu commitversion** command to continue with the eFSU. To continue, enter the **issu commitversion** command to upgrade the VSS standby chassis and complete the eFSU sequence. The VSS standby chassis reboots, reloads with the new image, and initializes as the VSS standby chassis in the SSO redundancy state, running the new image. This step is complete when the chassis configuration is synchronized, as indicated by the “Bulk sync succeeded” message.

```
Router# issu commitversion
Building configuration...
[OK]
000148: Aug  6 17:17:28.267 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet2/7/4, changed state to down
000149: Aug  6 17:17:28.287 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/2/4, changed state to down
```

(Deleted many interface and protocol down messages)

```
%issu commitversion executed successfully
```

(Deleted many interface and protocol down messages, then interface and protocol up messages)

```
000181: Aug  6 17:41:51.086 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/2/5, changed state to up
000182: Aug  6 17:42:52.290 PST: %HA_CONFIG_SYNC-6-BULK_CFGSYNC_SUCCEED: Bulk Sync
succeeded
```

Verify That the Upgrade is Complete

You can now enter the **show issu state detail** command and the **show redundancy** command to check the results of the eFSU. In this example, both chassis are now running the “newversion” image, indicating that the eFSU was successful. Because the eFSU has completed, the two chassis will be once again in the ISSU Init Version state, as they were before the eFSU was initiated.

```
Router# show issu state detail
                          Slot = 2/7
                      RP State = Active
                    ISSU State = Init
                 Boot Variable =
disk0:s72033-newversion.v2,12;disk0:s72033-oldversion.v1,12
                Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:s72033-newversion.v2
                Variable Store = PrstVbl

                          Slot = 1/2
                      RP State = Standby
                    ISSU State = Init
                 Boot Variable =
disk0:s72033-newversion.v2,12;disk0:s72033-oldversion.v1,12
```

```
                Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:s72033-newversion.v2

Router# show redundancy status
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Primary
        Unit ID = 39

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
 Communications = Up

   client count = 134
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 18
           RF debug mask = 0x0
```

Tip

For additional information about Cisco Catalyst 6500 series switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html