Virtual Switching Systems

Note

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

  http://www.cisco.com/en/US/products/ps11846/prod_command_reference_list.html
- Cisco IOS Release 15.2SY supports only Ethernet interfaces. Cisco IOS Release 15.2SY does not support any WAN features or commands.

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

Participate in the Technical Documentation Ideas forum

Prerequisites for VSS

The VSS configurations in the startup-config file must match on both chassis.

Restrictions for VSS

General VSS Restrictions

- VSS mode does not support supervisor engine redundancy within a 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.
- Out-of-band MAC address table synchronization among DFC-equipped switching modules (the **mac address-table synchronize** command) is enabled automatically in VSS mode, which is the recommended configuration.
- Because the output of the **show running-config** command on ICS supervisor engines could be out of sync with the active supervisor engine, ICS supervisor engines do not support the **show running-config** command. The active and standby supervisor engines support the **show running-config** command.

VSL 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 platform 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 (MEC) Restrictions

- All links in an MEC must terminate locally on the active or 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 "Information about LACP 1:1 Redundancy" section on page 23-6.
- An MEC can be connected to another MEC in a different VSS domain.
- Ports on the supervisor engines are not stateful and will experience a reset across switchovers (see the "Switchover Process Restrictions" section on page 7-2).
- With an MEC that includes supervisor engine ports configured on a VSS that supports the VSS Quad-Sup SSO (VS4O) feature, be aware of CSCuh51005.

Dual-Active Detection Restrictions

- If Flex Links are configured on the VSS, use PAgP dual-active detection.
- 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 Chapter 62, "Restrictions for PFC QoS."

VSS Mode Service Module Restrictions

- When configuring and attaching VLAN groups to a service module interface, 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, use the **switch** {**1** | **2**} command keyword.
- EtherChannel load balancing (ECLB) is not supported between an IDSM-2 in the active chassis and an IDSM-2 in the 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 VSS mode, see the configuration guide and command reference for the service module.

Information About Virtual Switching Systems

VSS Overview

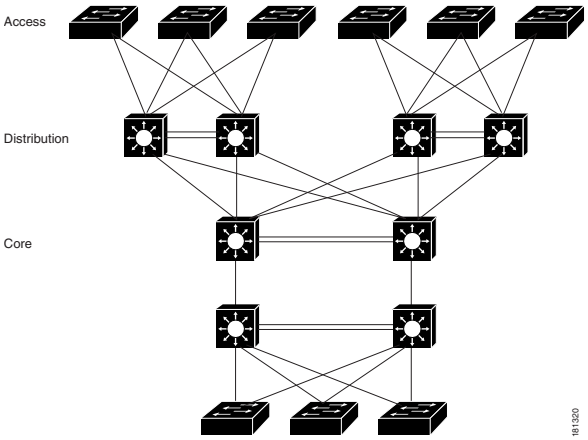
VSS Topology

Network operators increase network reliability by configuring switches in redundant pairs and by provisioning links to both switches in the redundant pair. Figure 11-1 shows a typical 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.

VSS mode combines a pair of switches into a single network element. VSS mode manages the redundant links, which externally act as a single port channel.

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

Figure 11-1 Typical Network Design



Key Concepts

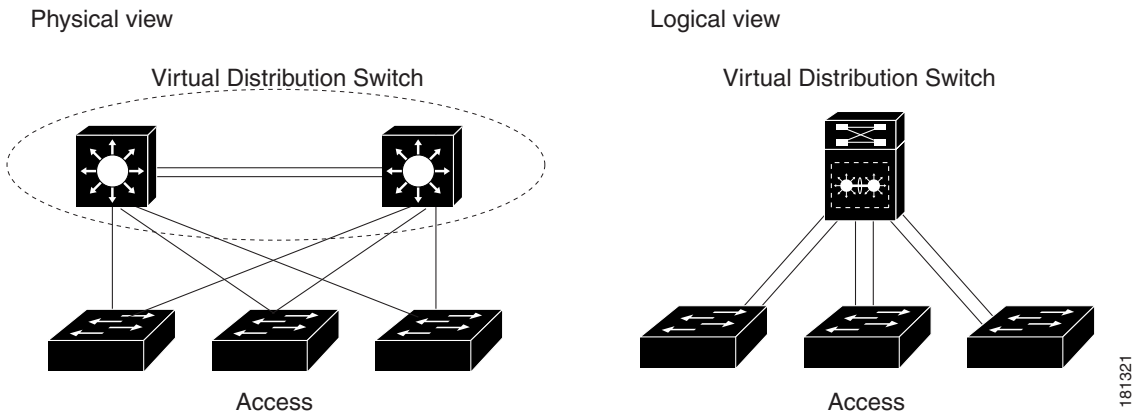
- Virtual Switching System, page 11-5
- Active and Standby Chassis, page 11-6
- Virtual Switch Link, page 11-7
- Multichassis EtherChannel (MEC), page 11-7

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 11-2.

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

Figure 11-2 VSS in the Distribution Network



Active and Standby Chassis

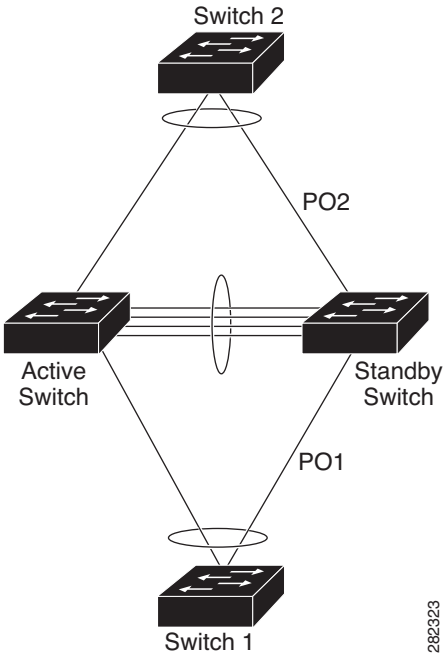
When you create or restart a VSS, the peer chassis negotiate their roles. One chassis becomes the active chassis, and the other chassis becomes the standby.

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

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

You can defer the traffic load on a multichassis EtherChannel (MEC) chassis to address traffic recovery performance during the standby chassis startup. For example, Figure 11-3 represents network layout where a VSS (active and standby switches) is interacting with an upstream switch (switch 2) and a downstream switch (switch 1).

Figure 11-3 Switch Interconnected Through VSS

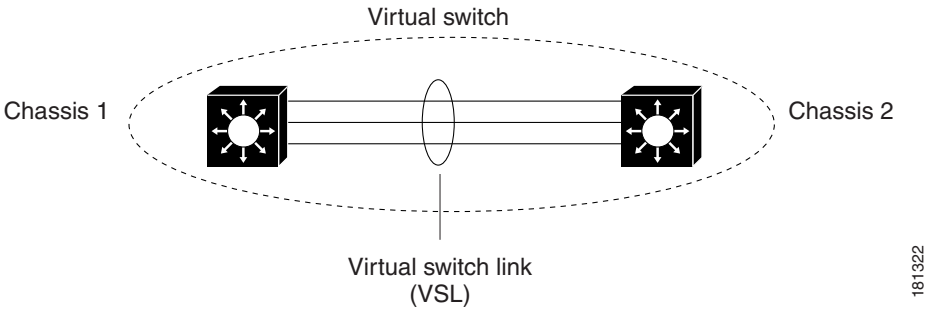


Virtual Switch Link

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 11-4. 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 11-4 Virtual Switch Link



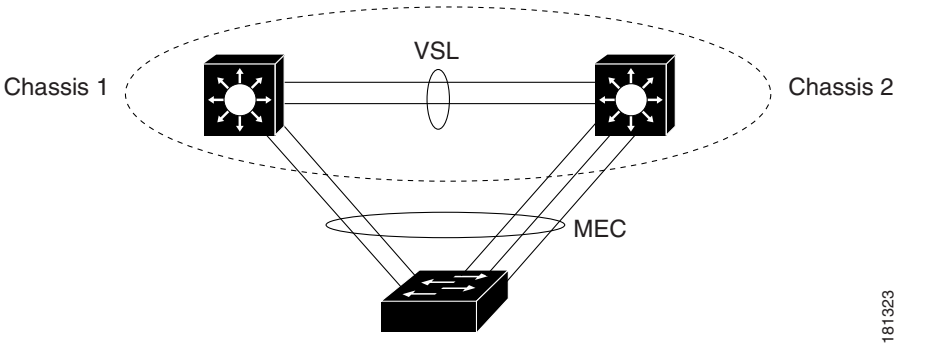
Multichassis EtherChannel (MEC)

An EtherChannel, which is configured on a port channel interface, is 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 MEC is a port channel with member ports on both chassis of the VSS. A connected non-VSS device views the MEC as a standard EtherChannel. See Figure 11-5.

VSS mode supports a maximum of 512 EtherChannels. This limit applies to the combined total of regular EtherChannels and MECs. Because the VSL requires two EtherChannel numbers (one for each chassis), there are 510 user-configurable EtherChannels. Service modules that use an internal EtherChannel are included in the total.

Figure 11-5 VSS with MEC



Note Ports on the supervisor engines are not stateful and will experience a reset across switchovers (see the "Switchover Process Restrictions" section on page 7-2).

VSS Functionality

Redundancy and High Availability

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

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

Note Ports on the supervisor engines are not stateful and will experience a reset across switchovers (see the "Switchover Process Restrictions" section on page 7-2).

If the VSL fails completely, the standby chassis assumes that the active chassis has failed, and initiates a switchover. After the switchover, if both chassis are 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 11-24.

Packet Handling

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

System Management

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

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

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

VSS Quad-Sup SSO (VS4O)

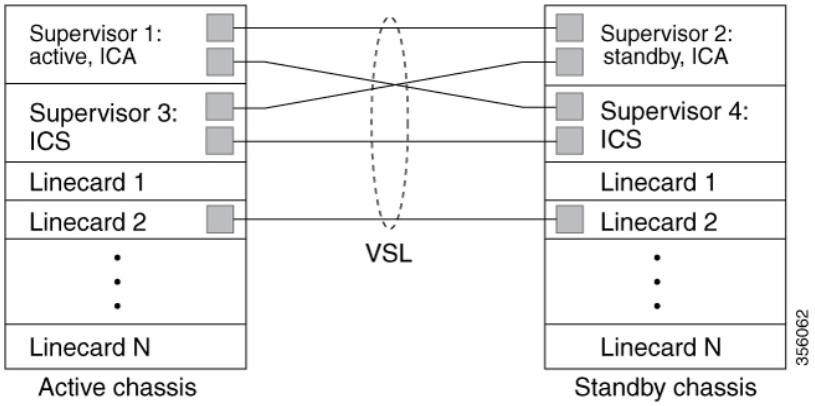
Release 15.1(1)SY1 and later releases support the VSS Quad-Sup SSO (VS4O) feature.

Best Practice for VSL Links Between Supervisors

If there is a VSL link from ICA to ICA supervisor engine, ICS to ICS supervisor engine, and line card to line card, the control link will always be from the line card. The next preferred control link is from ICS to ICS VSL link, and the least preferred control link is from ICA to ICA VSL link.

In case of an ISSU image upgrade, if there is no cross VSL link, the control link will always be from the ICS supervisor engine. If there is a cross VSL link, the control link can be from any of the ICA supervisor engines.

Figure 11-6 Typical VSS Quad-Supervisor Configuration



These are the quad-supervisor VSS roles:

- In-chassis active (ICA) supervisor engines—The VSS active supervisor engine in one chassis and the VSS standby supervisor engine in the other chassis are ICA supervisor engines.

  If the VSS active ICA supervisor engine crashes, a switchover to the standby ICA supervisor engine in other chassis occurs. Both VSS chassis remain active. All switching modules remain active.

  In the chassis with the previously active ICA supervisor engine, an SSO switchover from the previously active ICA supervisor engine to the ICS standby supervisor engine occurs and the ICS takes over as the ICA. The failed ICA reloads and becomes the ICS.

  You can verify the switchover mode of the supervisor engines by entering the **show module** command.

- In-chassis standby (ICS) supervisor engines—The other supervisor engines are ICS supervisor engines. The supervisor engine uplinks ports are available to forward traffic.

Note ICS supervisor engines do not support **show** commands. To avoid tracebacks, do not issue **show** commands on ICS supervisor engines.

If the supervisor engine PFC modes do not match then an ICS supervisor engine is reset to ROMMON. Configure both chassis to run in the same PFC mode.

ICS supervisor engines boot in SSO standby mode, which fully initializes and configures the ICS and maintains stateful feature and user session information.

To verify the switchover mode of the supervisor engines, enter the **show module** command.

When not in VSS quad supervisor engine mode, if you insert a supervisor engine to be an ICS , the supervisor engine resets to update the supervisor engine number and then reboots before going online.

Quad-supervisor SSO (VS4O) supports eFSU upgrades. You can upgrade or downgrade your VSS system using ISSU. See "How to Upgrade a VSS" section on page 11-56 for more information about eFSU upgrades.

Interface Naming Convention

In VSS mode, interfaces are specified using 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, VSS mode has feature parity with non-VSS mode. Major exceptions include:

- VSS mode does not support supervisor engine redundancy within a chassis.
- Port-based QoS and PACLs can be applied to any physical port, except VSL ports. PACLs can be applied to no more than 2,046 ports.

Hardware Requirements

Chassis and Modules

Table 11-1 VSS Hardware Requirements

Hardware	Count	Requirements
Chassis	2	All chassis supported with Supervisor Engine 2T in Cisco IOS Release 15.2SY support VSS mode. **Note** .VSS is supported only between two chassis of Catalyst 6500 Series Switches or between two chassis of Catalyst 6800 Series Switches, but not between a chassis of Catalyst 6500 Series Switch and a chassis of Catalyst 6800 Series Switch. The two chassis need not be identical as long as they both belong to the same series switches.
Supervisor Engines	2	Either two VS-SUP2T-10G or two VS-SUP2T-10G-XL supervisor engines. The two supervisor engines must match exactly.
Switching Modules	2+	VSS mode support as shown in the Release Notes. In VSS mode, unsupported switching modules remain powered off.

VSL Hardware Requirements

The VSL EtherChannel supports only 40-Gigabit and 10-Gigabit Ethernet ports. The ports can be located on the supervisor engine (recommended) or on one of the following switching modules:

- WS-X6904-40G-2T
- WS-X6908-10GE
- WS-X6816-10T-2T, WS-X6716-10T
- WS-X6816-10G-2T, WS-X6716-10G
- C6800-16P10G

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 40-Gigabit or 10-Gigabit Ethernet ports on switching modules that support the VSL.

Note

- When using the ports on a switching module that can operate in oversubscribed mode as VSL links, you must operate the ports in performance mode, not in oversubscription mode. Enter the **no hw-module switch x slot y oversubscription port-group** *num* command when configuring the switching module. If you enter the **no hw-module switch** *switch_number* **slot** *slot_number* **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 mode for VSL with the unused ports administratively shutdown, while the others can still operate in oversubscribed mode.

PFC, DFC, and CFC Requirements

Switching modules with a CFC, DFC4, or DFC4XL support VSS mode.

With a PFC4, the VSS will automatically operate in PFC4 mode, even if some of the modules have a DFC4XL. With a PFC4XL, but some modules equipped with a DFC4, you need to configure the VSS to operate in PFC4 mode. The **platform hardware vsl pfc mode non-xl** configuration command sets the system to operate in PFC4 mode after the next restart. See the "SSO Dependencies" section on page 11-27 for further details about this command.

Multichassis EtherChannel Requirements

Physical links from any module with a CFC, DFC4, or DFC4XL can be used to implement a Multichassis EtherChannel (MEC).

Service Module Support

- Application Control Engine (ACE):
  - ACE20-MOD-K9
  - ACE30-MOD-K9
- ASA Services Module: WS-SVC-ASA-SM1-K9
- Firewall Services Module (FWSM): WS-SVC-FWM-1-K9
- Network Analysis Module (NAM):
  - WS-SVC-NAM-1
  - WS-SVC-NAM-2
  - WS-SVC-NAM3-6G-K9
- Wireless Services Module (WiSM):
  - WS-SVC-WISM-1-K9
  - WS-SVC-WISM2

Note Before deploying a service module in VSS mode, upgrade the module to the minimum supported release in standalone mode. See the service module release notes for information about the minimum required service module software version.

Information about VSL Topology

A VSS is two chassis that communicate using the VSL, which is a special port group. 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 40- or 10-Gigabit Ethernet ports. This configuration provides additional VSL capacity. See Figure 11-7 for an example topology.

Figure 11-7 VSL Topology Example

VSS Redundancy

Overview

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

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

If the active supervisor engine fails, the standby supervisor engine initiates a switchover and assumes the active role.

RPR and SSO Redundancy

The VSS normally runs stateful switchover (SSO) between the active and standby supervisor engines (see Figure 11-8). The VSS determines the role of each supervisor engine during initialization.

Figure 11-8 Chassis Roles in VSS Mode

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

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

- Both supervisor engines are running the same software version.
- The VSL-related configuration in the two chassis matches.
- The PFC mode matches.
- SSO and nonstop forwarding (NSF) are configured on both chassis.

See the "SSO Dependencies" section on page 11-27 for additional details about the requirements for SSO redundancy on a VSS. See Chapter 8, "Nonstop Forwarding (NSF)" for information about configuring SSO and NSF.

With SSO redundancy, the supervisor engine in the standby chassis runs in hot standby state and is always ready to assume control following a fault on the active supervisor engine. Configuration, forwarding, and state information are synchronized from the active supervisor engine to the redundant supervisor engine at startup and whenever changes to the 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 uses route processor redundancy (RPR). In RPR mode, the active supervisor engine does not synchronize configuration changes or state information with the standby. The standby supervisor engine is only partially initialized and the switching modules on the standby supervisor are not powered up. If a switchover occurs, the standby supervisor engine completes its initialization and powers up the switching modules. Traffic is disrupted for approximately 2 minutes.

Failed Chassis Recovery

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

If the 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 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 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 standby chassis become operational after the SSO.

After the SSO, much of the processing power of the active supervisor engine is consumed in bringing up a large number of ports simultaneously in the 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:

- 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 Standby Recovery" section on page 11-47.
- 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 11-18 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 standby chassis detects complete VSL link failure, it initiates a stateful switchover (SSO). If the 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 active chassis is still operational, this is a dual-active scenario. The VSS detects that both chassis are operating in active mode and performs recovery action. See the "Dual-Active Detection" section on page 11-24 for additional details about the dual-active scenario.

User Actions

From the 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 standby chassis, use **redundancy reload peer** command.

To force a switchover from the active to the 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

Overview

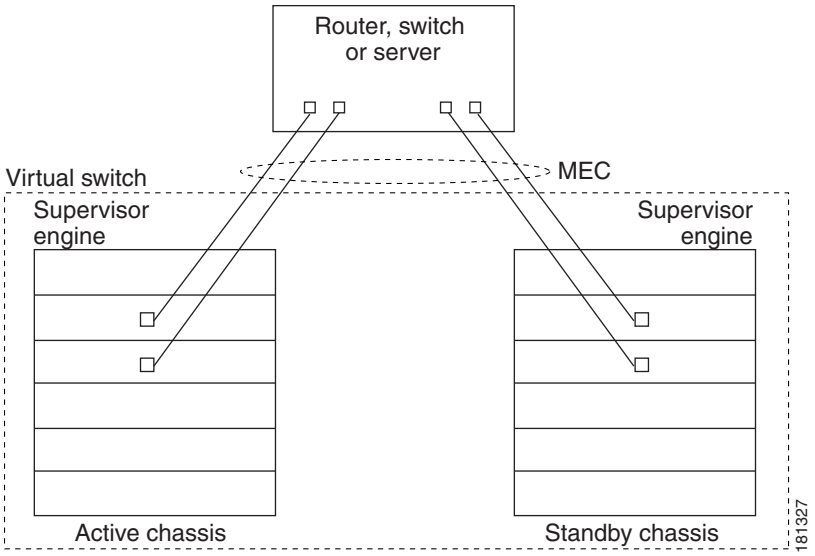
A multichassis EtherChannel is an EtherChannel with ports that terminate on both chassis of the VSS (see Figure 11-9). A VSS MEC can connect to any network element that supports EtherChannel (such as a host, server, router, or switch).

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 active chassis, the VSS will select an MEC link from the 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 active chassis. PAgP or LACP control packets destined for an MEC link on the standby chassis are sent across VSL.

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

Figure 11-9 MEC Topology



MEC Failure Scenarios

Note 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).

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 Active Chassis Fail

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

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

All MEC Links to the Standby Chassis Fail

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

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

Standby Chassis Failure

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

Active Chassis Failure

Active chassis failure results in a stateful switchover (SSO). See the "VSS Redundancy" section on page 11-13 for details about SSO on a VSS. After the switchover, the MEC is operational on the new 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 active chassis.

Failed Chassis MEC Recovery

When a failed chassis returns to service as the new 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 11-49 for details about configuring port load share deferral.

Packet Handling

Packet Handling Overview

In VSS mode, the 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. VSS mode minimizes the amount of data traffic that must traverse the VSL.

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 active supervisor engine, but received or transmitted by interfaces on the standby chassis. Control traffic also includes module programming between the active supervisor engine and switching modules on the 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 active supervisor engine where the ingress interface is on the 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 standby chassis to the 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

- Layer 2 Protocol Overview, page 11-20
- Spanning Tree Protocol, page 11-20
- Virtual Trunk Protocol, page 11-20
- EtherChannel Control Protocols, page 11-20
- Multicast Protocols, page 11-20

Layer 2 Protocol Overview

The 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 standby chassis switching modules must traverse the VSL to reach the active supervisor engine.

Spanning Tree Protocol

The active chassis runs Spanning Tree Protocol (STP). The standby chassis redirects STP BPDUs across the VSL to the 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 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 11-24.

Multicast Protocols

With Release 15.1(1)SY1 and later releases, fast-redirect optimization makes multicast traffic redirection between inter-chassis or intra-chassis line cards faster for Layer 2 trunk and Layer3 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

Layer 3 Protocol Overview

The RP on the 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 standby chassis is actively forwarding traffic, the active supervisor engine distributes updates to the standby supervisor engine PFC and all standby chassis DFCs.

IPv4

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

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

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

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

The supervisor engine on the 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 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. See Chapter 8, "Nonstop Forwarding (NSF)."

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, MPLS, and VPLS

The VSS supports IPv6 unicast, MPLS, and VPLS.

IPv4 Multicast

The IPv4 multicast protocols run on the active supervisor engine. Internet Group Management Protocol (IGMP) and Protocol Independent Multicast (PIM) protocol packets received on the standby supervisor engine are transmitted across VSL to the active chassis.

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

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

For Layer 3 multicast in the VSS, learned multicast routes are stored in hardware in the 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 active chassis does not program the multicast expansion table (MET) on the standby chassis. The standby supervisor engine programs the outgoing interface hardware entries for all local multicast receivers

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

In egress replication 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 active supervisor engine. Incoming packets to the 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 active supervisor engine, the standby supervisor engine, and all DFCs.

SPAN Support with VSS

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

- VSL ports cannot be a SPAN destination.
- VSL ports cannot be an RSPAN, ERSPAN, or egress-only SPAN source.
- If a VSL port is configured as a local SPAN source, the SPAN destination interface must be on the same chassis as the source interface.
- SPAN copies are always made on the chassis where the ingress port is located.
- Two VSLs cannot share the same SPAN session.
- A pair of LTL indices are used to avoid duplicate SPAN copies across VSL interfaces.

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

With a VSL port as a SPAN source, the following limitations apply:

- The SPAN destination must be on the same chassis.
- Port channel interfaces cannot be the SPAN destination.

System Monitoring

Power Management

You can control power-related functions for the standby chassis from the active chassis. For example, use the **(no) power enable switch** command to control power to the modules and slots on the 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 standby chassis reports notifications to the active supervisor engine. The active chassis gathers log messages for both chassis. The active chassis synchronizes the calendar and system clock to the standby chassis.

File System Access

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

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.

VSL Diagnostics

The following VSL-specific diagnostics tests are disruptive:

- TestVSActiveToStandbyLoopback
- TestVslBridgeLink
- TestVslLocalLoopback

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

- TestVslStatus

Service Modules

The following system monitoring and system management guidelines apply to service modules supported in VSS mode:

- The supervisor engine in the same chassis as the service module controls service module power up. After service modules are online, you can initiate sessions from the active supervisor engine to the service module.
- Use the **session** command to connect to a service module. If a service module is in the standby chassis, the session runs over the VSL.
- The active chassis performs graceful shutdown of all service modules, including any in the standby chassis.

Network Management

Telnet over SSH Sessions and the Web Browser User Interface

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

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

A VSS switchover disconnects Telnet over SSH sessions and web browser sessions.

SNMP

The SNMP agent runs on the active supervisor engine. CISCO-VIRTUAL-SWITCH-MIB is the MIB for VSS 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

Console Connections

Connect console cables to both supervisor engine console ports. The console on the standby chassis adds the characters “-stdby” to the command line prompt to indicate that the chassis is operating in standby mode. You cannot enter configuration mode on the standby chassis console.

The following example shows the prompt on the 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

Dual-Active Detection Overview

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

If the original active chassis is still operational, both chassis are now 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 two 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.
- dual-active fast-hello—Uses special hello messages over a backup Ethernet connection.

You can configure both detection methods to be 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.

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 active chassis is still operational, it detects the dual-active scenario because the active ID in the PAgP messages changes. This chassis initiates recovery actions as described in the "Recovery Actions" section on page 11-26.

Dual-Active Detection Using Dual-Active Fast Hello Packets

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 11-26. 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. For more information, see the "Configuring Enhanced PAgP Dual-Active Detection" section on page 11-49.

Recovery Actions

An 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 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

VSS Initialization Overview

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 active role. The VSS forms when the second chassis becomes operational and both chassis bring up their VSL interfaces.

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 (active or 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 standby chassis sends virtual switch information from the startup-config file to the active chassis. The 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 active chassis console and the 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 active chassis, and then restart the standby chassis.
- PFC mode check—If both supervisor engines are provisioned with PFC4, the VSS will automatically operate in PFC4 mode, even if some of the switching modules are equipped with DFC4XLs.

  However, if the supervisor engines are provisioned with PFC4XL and there is a mixture of DFC4 and DFC4XL switching modules, the system PFC mode will depend on how the DFC4XL and DFC4XL 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 PFC4XL and all the switching modules in the chassis are provisioned with DFC4XL, the PFC mode for the chassis is PFC4XL. However, if any of the switching modules is provisioned with DFC4, the chassis PFC mode will be set to PFC4. 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 non-xl** command to force the VSS to operate in PFC4 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 8, “Nonstop Forwarding (NSF).”

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 11-13.

Initialization Procedure

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 (active or 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 (active or standby).
3. The active chassis completes the boot sequence, including the consistency check described in the "SSO Dependencies" section on page 11-27.
4. If the consistency check completed successfully, the standby chassis comes up in SSO standby mode. If the consistency check failed, the standby chassis comes up in RPR mode.
5. The active chassis synchronizes configuration and application data to the standby chassis.

System Initialization

If you boot both chassis simultaneously, the VSL ports become active, and the chassis will come up as active and 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 active. When you subsequently boot up the other chassis, the VSL links become active, and the new chassis comes up as 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 active and the other chassis initiates recovery from the dual-active scenario. For further information, see the "Configuring Dual-Active Detection" section on page 11-49.

Default Settings for VSS

None.

How to Configure a VSS

Configuring Easy VSS

Easy VSS mode allows the conversion of standalone switches into one virtual switching system (VSS) using a single command.

The following prerequisites must be fulfilled for successful configuration of Easy VSS:

- Both the switches must be in standalone mode
- Easy VSS mode must be enabled on both the switches
- CDP should be running between interfaces

Note

- Any manual configurations done using **switch virtual domain** *domain id* command will be lost.
- On converting a standalone switch into VSS, the switch boots with the system image (image version displayed in the output of **show version** command) and not the image specified in the boot variable.

To configure Easy VSS mode, perform the following tasks on switch 1:

Command	Purpose
`Switch (config)#` **switch virtual easy**	Enables easy VSS feature, you must enable this feature on both the switches.
`Switch (config)#` **switch convert mode easy-virtual-switch links** *<interface1> [interface2.....interface8]* **[with-vsl-encryption|[domain<***id***>]**	The switch where this command is executed becomes switch 1. List of interfaces is populated using CDP protocol. You can verify the list of interfaces using *switch convert mode easy-virtual-switch links?* command

To configure Easy VSS mode, perform the following tasks on switch 2:

Command	Purpose
`Switch (config)#` **switch virtual easy**	Enables easy VSS feature, you must enable this feature on both the switches.

Easy VSS using with-vsl-encryption

For using with-vsl-encryption option, pairwaise master key (PMK) needs to be configured on both the switches.
To configure Easy VSS with VSL encryption, perform the following tasks on switch 1:

Command	Purpose
`Switch (config)#` **switch virtual easy**	Enables easy VSS feature, you must enable this feature on both the switches.
`Switch (config)#` **end**	Executes the configuration.
`Switch#` **switch pmk** *value*	Configures PMK with the specified value.
`Switch#` **switch convert mode easy-virtual-switch links** *interface1 <interface2.....interface8>* **with-vsl-encryption** *domain<id>*	The switch where this command is executed becomes switch 1. List of interfaces is populated using CDP protocol. You can verify the list of interfaces using *switch convert mode easy-virtual-switch links?* command

To configure Easy VSS with VSL encryption, perform the following tasks on switch 2:

Command	Purpose
`Switch (config)#` **switch virtual easy**	Enables easy VSS feature, you must enable this feature on both the switches.
`Switch (config)#` **end**	Executes the configuration.
`Switch#` **switch pmk** *value*	Configures PMK with the specified value.

Verifying Easy VSS Configuration

This example shows how to verify that both switches are running in standalone mode:

```
Switch #show switch virtual
Switch Mode : Standalone
Not in Virtual Switch mode due to:
Domain ID is not configured
```

This example shows how to verify if the connected interfaces are up:

```
Switch #show interfaces tenGigabitEthernet 5/15
TenGigabitEthernet5/15 is up, line protocol is up (connected)
```

This example shows how to verify that CDP is running fine between interfaces:

```
Switch #show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
           S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
           D - Remote, C - CVTA, M - Two-port Mac Relay
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
Term2            Ten 5/15          148             R S I  C6880-X
Total cdp entries displayed : 1
```

This example shows the output of *show switch virtual* command:

```
Switch1 #show swi virtual
Switch mode                   : Virtual Switch
Virtual switch domain number : 2
Ten 5/15
   Local switch number          : 1
Local switch operational role: Virtual Switch Active
Peer switch number           : 2
Peer switch operational role : Virtual Switch Standby
```

This example shows the output of *show switch virtual link* command:

```
Switch1 #sh swi virtual link
VSL Status : UP
VSL Uptime : 2 minutes
VSL SCP Ping : Pass
VSL ICC Ping : Pass
VSL Control Link : Te1/5/15
VSL Encryption : Configured Mode - On Operational Mode - On ->VSL encryption has been
enabled
```

Converting to a VSS

- VSS Conversion Overview, page 11-31
- Backing Up the Standalone Configuration, page 11-32

VSS Conversion Overview

The standalone mode is the default operating mode (a single chassis switch). VSS mode combines two standalone switches into one virtual switching system (VSS), operating in VSS mode.

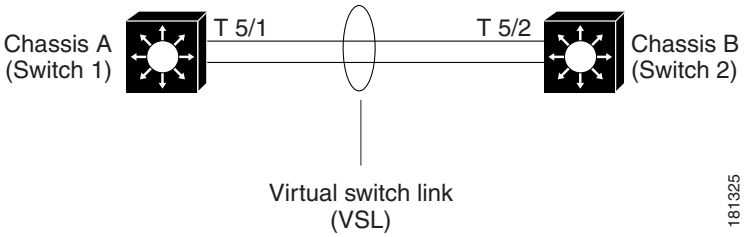
Note When you convert two standalone switches into one VSS, all non-VSL configuration settings on the standby chassis revert to default settings.

To convert two standalone chassis into a VSS, 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 the procedures that follow, the example commands assume the configuration shown in Figure 11-10.

Figure 11-10 Example VSS



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

Backing Up the Standalone Configuration

Save the configuration files for both chassis. These files are needed to revert to standalone mode from virtual switch mode.

Switch 1 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.

Switch 2 Task

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

Configuring SSO and NSF

SSO and NSF must be configured and enabled on both chassis.

Switch 1 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 ospf` *processID*	Enables an OSPF routing process, which places the router in router configuration mode.
Step 5	`Switch-1(config-router)# nsf`	Enables NSF operations for OSPF.
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.

Switch 2 Task

	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.
Step 3	`Switch-2(config-red)# exit`	Exits redundancy configuration mode.
Step 4	`Switch-2(config)# router ospf processID`	Enables an OSPF routing process, which places the router in router configuration mode.
Step 5	`Switch-2(config-router)# nsf`	Enables NSF operations for OSPF.
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 8, "Nonstop Forwarding (NSF)."

Assigning Virtual Switch Domain and Switch Numbers

Configure the same virtual switch domain number on both chassis. 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.

Switch 1 Task

	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.

Switch 2 Task

	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 the VSL Port Channel

The VSL is configured with a unique port channel on each chassis. During the conversion, the VSS configures both port channels on the active chassis. If the 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.
```

Switch 1 Task

	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.

Switch 2 Task

	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.

Configuring the VSL Ports

You must add the VSL physical ports to the port channel. In the following example, 10-Gigabit Ethernet ports 3/1 and 3/2 on Switch 1 are connected to 10-Gigabit Ethernet ports 5/2 and 5/3 on Switch 2. For VSL line redundancy, configure the VSL with at least two ports per chassis. For module redundancy, the two ports can be on different switching modules in each chassis.

Switch 1 Task

	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	`Switch-1(config-if)# no shutdown`	Activates the port.

Switch 2 Task

	Command	Purpose
Step 1	`Switch-2(config)# interface range tengigabitethernet 5/2-3`	Enters configuration mode for interface range tengigabitethernet 5/2-3 on Switch 2.
Step 2	`Switch-2(config-if)# channel-group 20 mode on`	Adds this interface to channel group 20.
Step 3	`Switch-2(config-if)# no shutdown`	Activates the port.

Verifying the PFC Operating Mode

Ensure that the PFC operating mode matches on both chassis. 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 PFC4XL mode, you can configure it to use PFC4 mode with the **platform hardware vsl pfc mode non-xl** command.

Switch 1 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-1(config)# platform hardware vsl pfc mode non-xl`	(Optional) Sets the PFC operating mode to PFC4 on Chassis A.

Switch 2 Task

	Command	Purpose
Step 3	`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 4	`Switch-2(config)# platform hardware vsl pfc mode non-xl`	(Optional) Sets the PFC operating mode to PFC4 on Chassis B.

Converting the Chassis to Virtual Switch Mode

Conversion to VSS 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_#*.

Before restarting, 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.

Switch 1 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.

Switch 2Task

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.

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_#*).

Auto-Configuring the Standby VSL Information

The two chassis now form a VSS, and the system will auto-configure the standby VSL. After the merge has completed successfully, enter all configuration commands for the VSS on the active chassis. The startup configuration file is automatically synchronized to the standby chassis after the standby chassis reaches the ready state. The VSSmode automatically merges the configuration information on the standby chassis.

All non-VSL interface configurations on the standby chassis revert to the default configuration and non-VSL related configurations are not merged. If you fail to perform any of the required configurations, you will have to repeat the configuration on the active chassis. Auto-configuration merges these commands for the standby chassis:

- **hw-module switch** *number* **slot** *number*
- **switch virtual domain** *number*
- **switch** *number* **priority** *priority*

- **power redundancy-mode combined switch** *number*
- **no power enable switch** *num* **module** *number*
- **interface port-channel** *num* **switch virtual link** *number*
- **interface** *type switch_#/slot_#/port_#* **channel-group** *number* **mode on**

(Optional) Configuring Standby Chassis Modules

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

Configurations for the standby chassis modules revert 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(50)SY 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

These commands display basic information about the VSS:

Command	Purpose
show switch virtual	Displays the virtual switch domain number, and the switch number and role for each of the chassis.
show switch virtual role	Displays the role, switch number, and priority for each of the chassis in the VSS.
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

Copying the VSS Configuration to a Backup File

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

When you convert the active chassis to standalone mode, the 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 standby chassis of the VSS becomes active. VSL links on this chassis are down because the peer is no longer available.

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

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 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 standby state, you can make it the 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.

Note If you make configuration changes to the switch priority, the changes only take effect after you save the running configuration to the startup configuration file and perform a reload. The **show switch virtual role** command shows the operating and configured priority values. You can manually set the standby switch to 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 the PFC Mode

If you have a mixture of DFC4 and DFC4XL switching modules in the VSS, set the PFC mode by performing this task:

Command	Purpose
`Router(config)# platform hardware vsl pfc mode non-xl`	Sets the PFC configuration mode for the VSS to PFC4. **Note** This command requires a system reload before it takes effect.

This example shows how to set the PFC configuration mode for the VSS to PFC4. You can wait until the next maintenance window to perform the **reload** command.

```
Router(config)# platform hardware vsl pfc mode non-xl
Router(config)# end
Router# reload
```

If all the supervisor engines and switching modules in the VSS are XL, the following warning is displayed if you set the PFC mode to PFC4:

```
Router(config)# platform hardware vsl pfc mode non-xl
PFC Preferred Mode: PFC4XL. The discrepancy between Operating Mode and
Preferred Mode could be due to PFC mode config. Your System has all PFC4XL modules.
Remove ' platform hardware vsl pfc mode non-xl ' from global config.
```

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

```
Router# show platform hardware pfc mode
PFC operating mode : PFC4
Configured PFC operating mode : PFC4
```

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
```

Configuring VSL Encryption

VSL Encryption Overview

Cisco IOS Release 15.2SY supports HW-based encryption on a VSL configured on a Supervisor Engine 2T or WS-X6908-10GE switching module. VSL encryption uses an encryption key that you manually configure. The encryption key is stored securely.

VSL Encryption Restrictions

- VSL encryption requires a MACSec license on each chassis.
- The chassis must be rebooted to configure an encryption key or enable VSL encryption.
- You enter the encryption key on the active chassis. You cannot enter the encryption key on the standby chassis.
- If it is acceptable to send the key as plain text over the VSL to the other chassis, then you can allow one chassis to send the key to the other chassis. For maximum security, configure the encryption key on each chassis.
- There are no **show** commands that display the encryption key.
- You cannot remove the encryption key while VSL encryption is enabled.
- The following commands take effect after a reboot:
  - To remove the encryption key, enter the **clear switch pmk** EXEC mode command.
  - To disable VSL encryption, enter the **no vsl-encryption** virtual switch domain configuration submode command.
- If the encryption key and VSL encryption state on the two chassis do not match, the VSL does not transition to the link-up state.

- In VSS mode, you cannot configure the FIPS encryption mode without VSL encryption. To avoid a system shutdown, enable VSL encryption before you enable FIPS encryption mode. (CSCts96040, CSCtx58304)

Configuring the VSL Encryption Key

To configure the VSL encryption key, perform this task:

Command	Purpose
`Router# switch pmk encryption_key`	Configures the VSL encryption key. • *encryption_key* is a hexadecimal string up to 32 characters (256 bits). • You will be asked if you want to automatically synchronize the encryption key. If you do not automatically synchronize the encryption key, configure the same encryption key on the other chassis.

This example show how to configure a VSL encryption key:

```
Router# switch pmk encryption_key
Key effective only upon reboot and will override old VSL PMK.
Key needs to be provisioned on both VSS switches.
Warning - Sending the key to standby will cause the key to be sent over an unencrypted VSL
link.
Do you want to automatically synchronize the key [yes/no]?
```

Enabling VSL Encryption

To enable VSL encryption, perform this task:

	Command	Purpose
Step 1	`Router(config)# switch virtual domain domain_id`	Enters VSS configuration mode.
Step 2	`Router(config-vs-domain)# vsl-encryption`	Enables VSL encryption.

This example shows how to enable VSL encryption:

```
Router(config)# switch virtual domain domain_id
Router(config-vs-domain)# vsl-encryption
```

Note

- If you manually configure the encryption key on each chassis:
  - Configure the encryption key on the active chassis.
  - Reboot the active chassis; the standby chassis becomes active.
  - Configure the encryption key on new active chassis.
  - Enable VSL encryption.
  - Reboot the entire VSS.
- If you allow the encryption key to be sent to the standby chassis, reboot the entire VSS after configuring the encryption key and enabling VSL encryption on the active chassis.

Displaying the VSL Encryption State

This example shows how to display the VSL encryption state:

```
Router# show switch virtual link | include Encryption
VSL Encryption : Configured Mode - On, Operational Mode - On
```

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 **platform qos 10g-only** global configuration command.

In Cisco IOS Release 12.2(50)SY and later releases, when the **platform 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 11-2 describes the available interface subcommands.

Table 11-2 Interface Subcommands for VSL Port Channels

Subcommand	Description
default	Sets a command to its defaults.
description	Enters a text description for the interface.

Table 11-2 Interface Subcommands for VSL Port Channels (continued)

Subcommand	Description
exit	Exits from interface configuration mode.
load-interval	Specifies interval for load calculation for an interface.
logging	Configures logging for interface.
platform	Specifies platform-specific command.
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 11-3 describes the available interface subcommands.

Table 11-3 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 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 active chassis as the router MAC address.

In the rare case where both chassis later become inactive and then start up with the second supervisor engine becoming the initial 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 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 Standby Recovery

Instead of allowing all ports to be activated simultaneously when a failed chassis is restarted as the 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
`Router(config)# switch virtual domain 1`	Enters VSS configuration mode.
`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.
`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(50)SY support a maximum of 128 port channels.

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 time`	(Optional) Configures the port load share deferral interval for all port channels. • *time*—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.
```

Note To provide the best support for multicast traffic, configure the load share deferral feature on all EtherChannels that have member ports on more than one module.

Configuring Dual-Active Detection

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 below).

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 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 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.

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 switch/slot/port*	Specifies an interface to exclude from shutting down in dual-active recovery.

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 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 [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
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 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

- Opening a Session with a Service Module in a VSS, page 11-54
- Assigning a VLAN Group to a Firewall Service Module in a VSS, page 11-54
- Assigning a VLAN Group to an ACE Service Module in a VSS, page 11-55
- Verifying Injected Routes in a Service Module in a VSS, page 11-55

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

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:

```
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    16  CEF720 16 port 10GE with DFC           WS-X6716-10GE      SAL1215M55F
  3     1  Application Control Engine Module      ACE20-MOD-K9       SAD120603SU
.
.
.
```

How to Upgrade a VSS

Performing a Fast Software Upgrade of a VSS

The FSU of a VSS is similar to the RPR-based standalone chassis FSU described in Chapter 6, "Fast Software Upgrade." While the standalone chassis upgrade is initiated by reloading the standby supervisor engine, the VSS upgrade is initiated by reloading the standby chassis. During the FSU procedure, a software version mismatch between the active and the 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.

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 active and 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.
Step 8	`Router# redundancy reload peer`	Reloads the 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 standby chassis will be in RPR redundancy mode. **Note** Before reloading the standby chassis, make sure you wait long enough to ensure that all configuration synchronization changes have completed. **Note** When you perform FSU on a quad-supervisor VSS, you must run the **redundancy reload peer command** twice. In case of FSU on a dual-supervisor VSS, run the **redundancy reload peer** command only once.
Step 9	`Router# redundancy force-switchover`	Forces the standby chassis to assume the role of the active chassis running the new Cisco IOS image. The modules are reloaded and the module software is downloaded from the new active chassis. The old active chassis reboots with the new image and becomes the 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 Chapter 5, "Enhanced Fast Software Upgrade," 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

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

- There must be atleast one VSL link between supervisor engine in each chassis.
- Images from different features sets, regardless of release, fail the eFSU compatibility check.
- 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 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 Chapter 5, "Enhanced Fast Software Upgrade," 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.

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 11-59
- Loadversion Stage, page 11-59
- Runversion Stage, page 11-59
- Acceptversion Stage (Optional), page 11-60
- Commitversion Stage, page 11-60
- Abortversion (Optional), page 11-60

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 11-61
- Performing an eFSU Upgrade, page 11-61
- Aborting an eFSU Upgrade, page 11-62

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 11-62.

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
```

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]
```
(Deleted many lines)

```
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

Participate in the Technical Documentation Ideas forum