



Configuring Supervisor Engine Redundancy Using RPR and SSO on Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E

Catalyst 4500 series switches allow a standby supervisor engine to take over if the active supervisor engine fails. In software, supervisor engine redundancy is enabled by running the redundant supervisor engine in route processor redundancy (RPR) or stateful switchover (SSO) operating mode.



Note

The minimum ROMMON requirement for running SSO and RPR is 15.0(1r)SG1. VSS default mode is SSO and is not configurable. VSS does not support RPR mode.

This chapter describes how to configure supervisor engine redundancy on the Catalyst 4507R and Catalyst 4510R switches.



Note

For information on Cisco nonstop forwarding (NSF) with SSO, see [Chapter 13, “Configuring Cisco NSF with SSO Supervisor Engine Redundancy.”](#)

This chapter contains these major sections:

- [About Supervisor Engine Redundancy, page 12-2](#)
- [About Supervisor Engine Redundancy Synchronization, page 12-5](#)
- [Supervisor Engine Redundancy Guidelines and Restrictions, page 12-5](#)
- [Configuring Supervisor Engine Redundancy, page 12-7](#)
- [Performing a Manual Switchover, page 12-12](#)
- [Performing a Software Upgrade, page 12-12](#)
- [Manipulating Bootflash on the Standby Supervisor Engine, page 12-14](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Supervisor Engine Redundancy

These sections describe supervisor engine redundancy:

- [Overview, page 12-2](#)
- [RPR Operation, page 12-3](#)
- [SSO Operation, page 12-3](#)

Overview

With supervisor engine redundancy enabled, if the active supervisor engine fails or if a manual switchover is performed, the standby supervisor engine becomes the “new” active supervisor engine. The standby supervisor engine has been automatically initialized with the startup configuration of the active supervisor engine, shortening the switchover time (30 seconds or longer in RPR mode, depending on the configuration; subsecond in SSO mode).

In addition to the reduced switchover time, supervisor engine redundancy supports the following:

- Online insertion and removal (OIR) of the supervisor engine.
Supervisor engine redundancy allows OIR of the redundant supervisor engine for maintenance. When the redundant supervisor engine is inserted, the active supervisor engine detects its presence, and the redundant supervisor engine boots into a partially-initialized state in RPR mode and a fully-initialized state in SSO mode.
- Software upgrade. (See the “[Performing a Software Upgrade](#)” section on page 12-12.)
To minimize down time during software changes on the supervisor engine, load the new image on the standby supervisor engine, and conduct a switchover.

When power is first applied to a switch, the supervisor engine that boots first becomes the active supervisor engine and remains active until a switchover occurs.

A switchover will occur when one or more of the following events take place:

- The active supervisor engine fails (due to either hardware or software function) or is removed.
- A user forces a switchover.
- A user reloads the active supervisor engine.

[Table 12-1](#) provides information about chassis and supervisor engine support for redundancy.

Table 12-1 Chassis and Supervisor Support

Supported Switches	Supported Supervisor Engines
WS-C4507R-E, WS-C4510R-E, WS-C4507R+E, WS-C4510R+E	Supervisor Engine 9-E (WS-X45-SUP9-E), Supervisor Engine 8L-E (WS-X45-SUP8L-E), Supervisor Engine 8-E (WS-X45-SUP8-E), Supervisor Engine 7L-E (WS-X45-SUP7L-E), Supervisor Engine 7-E (WS-X45-SUP7-E),

RPR Operation

RPR is supported in Cisco IOS-XE Release 3.1.0SG and later releases. When a standby supervisor engine runs in RPR mode, it starts up in a partially-initialized state and is synchronized with the persistent configuration of the active supervisor engine.



Note

Persistent configuration includes the following components: startup-config, boot variables, config-register, and VLAN database.

The standby supervisor engine pauses the startup sequence after basic system initialization, and in the event that the active supervisor engine fails, the standby supervisor engine becomes the new active supervisor engine.

In a supervisor engine switchover, traffic is disrupted because in the RPR mode all of the physical ports restart since there is no state maintained between supervisor engines relating to module types and statuses. Upon switchover, when the standby supervisor engine completes its initialization, it will read hardware information directly from the module and become the active supervisor engine.

SSO Operation

SSO is supported in Cisco IOS-XE Release 3.1.0SG and later releases. When a standby supervisor engine runs in SSO mode, the standby supervisor engine starts up in a fully-initialized state and synchronizes with the persistent configuration and the running configuration of the active supervisor engine. It subsequently maintains the state on the protocols listed below, and all changes in hardware and software states for features that support stateful switchover are kept in synchronization. Consequently, it offers zero interruption to Layer 2 sessions in a redundant supervisor engine configuration.

Because the standby supervisor engine recognizes the hardware link status of every link, ports that were active before the switchover will remain active, including the uplink ports. However, because uplink ports are physically on the supervisor engine, they will be disconnected if the supervisor engine is removed.

If the active supervisor engine fails, the standby supervisor engine become active. This newly active supervisor engine uses existing Layer 2 switching information to continue forwarding traffic. Layer 3 forwarding will be delayed until the routing tables have been repopulated in the newly active supervisor engine.

After a switchover, Access Point (AP) connected ports take some time to converge. For faster STP convergence of AP connected ports, enable the **spanning-tree portfast edge** interface configuration command on all interfaces that are connected to APs. This enables the workstation or server to allow those devices to connect to the network immediately, rather than waiting for spanning tree to converge.

SSO supports stateful switchover of the following Layer 2 features.

The state of these features is preserved between both the active and standby supervisor engines:

- 802.3
- 802.3u
- 802.3x (Flow Control)
- 802.3ab (GE)
- 802.3z (Gigabit Ethernet including CWDM)

- 802.3ad (LACP)
- 802.1p (Layer 2 QoS)
- 802.1q
- 802.1X (Authentication)
- 802.1D (Spanning Tree Protocol)
- 802.3af (Inline power)
- PAgP
- VTP
- Dynamic ARP Inspection
- DHCP snooping
- IP source guard
- IGMP snooping (versions 1 and 2)
- DTP (802.1q and ISL)
- MST
- PVST+
- Rapid-PVST
- PortFast/UplinkFast/BackboneFast
- BPDU guard and filtering
- Voice VLAN
- Port security
- Unicast MAC filtering
- ACL (VACLs, PACLS, RACLS)
- QOS (DBL)
- Multicast storm control/broadcast storm control

SSO is compatible with the following list of features. However, the protocol database for these features is not synchronized between the standby and active supervisor engines:

- 802.1Q tunneling with Layer 2 Protocol Tunneling (L2PT)
- Baby giants
- Jumbo frame support
- CDP
- Flood blocking
- UDLD
- SPAN/RSPAN
- NetFlow

The following features are learned on the standby supervisor engine if the SSO feature is enabled:

- All Layer 3 protocols on Catalyst 4500 series switches (Switch Virtual Interfaces)

About Supervisor Engine Redundancy Synchronization

During normal operation, the persistent configuration (RPR and SSO) and the running configuration (SSO only) are synchronized by default between the two supervisor engines. In a switchover, the new active supervisor engine uses the current configuration.



Note You cannot enter CLI commands on the standby supervisor engine console.

These sections describe supervisor engine redundancy synchronization:

- [RPR Supervisor Engine Configuration Synchronization, page 12-5](#)
- [SSO Supervisor Engine Configuration Synchronization, page 12-5](#)

RPR Supervisor Engine Configuration Synchronization

Because the standby supervisor engine is only partially initialized in RPR mode, it interacts with the active supervisor engine only to receive configuration changes at startup and upon saving the configuration changes.

When a standby supervisor engine is running in RPR mode, the following events trigger synchronization of the configuration information:

- When the standby supervisor engine boots, the **auto-sync** command synchronizes the persistent configuration. This command is enabled by default. For details, refer to “[Synchronizing the Supervisor Engine Configurations](#)” section on page 12-10.
- When the active supervisor engine detects the standby supervisor engine, the configuration information is synchronized from the active supervisor engine to the standby supervisor engine. This synchronization overwrites any existing startup configuration file on the standby supervisor engine.
- When you make changes to the configuration, you must use the **write** command to save and synchronize the startup configuration to the standby supervisor engine.

SSO Supervisor Engine Configuration Synchronization

When a standby supervisor engine runs in SSO mode, the following events trigger synchronization of the configuration information:

- When the active supervisor detects the standby supervisor engine, synchronization of the persistent and running configuration takes place, allowing the standby supervisor engine to arrive at a fully-initiated state.
- When real-time changes occur, the active supervisor engine synchronizes the running-config and (or) the persistent configuration (if necessary) with the standby supervisor engine.
- When you change the configuration, you must use the **write** command to allow the active supervisor engine to save and synchronize the startup configuration to the standby supervisor engine.

Supervisor Engine Redundancy Guidelines and Restrictions

The following guidelines and restrictions apply to supervisor engine redundancy:

- If SSO mode cannot be established between the active and standby supervisor engines because of an incompatibility in the configuration file, a mismatched command list (MCL) is generated at the active supervisor engine and a reload into RPR mode is forced for the standby supervisor engine. Subsequent attempts to establish SSO, after removing the offending configuration and rebooting the standby supervisor engine with the exact same image, might cause the C4K_REDUNDANCY-2-`IOS_VERSION_CHECK_FAIL` and ISSU-3-`PEER_IMAGE_INCOMPATIBLE` messages to appear because the peer image is listed as incompatible. If the configuration problem can be corrected, you can clear the peer image from the incompatible list with the **redundancy config-sync ignore mismatched-commands EXEC** command while the peer is in a standby cold (RPR) state. This action allows the standby supervisor engine to boot in standby hot (SSO) state when it reloads.

Here are the steps:

-
- Step 1** Clear the offending configuration (that caused an MCL) while the standby supervisor engine is in standby cold (RPR) state.
- Step 2** Enter the **redundancy config-sync ignore mismatched-commands EXEC** command at the active standby supervisor engine.
- Step 3** Perform **write memory**.
- Step 4** Reload the standby supervisor engine with the **redundancy reload peer** command.
-

- RPR and SSO requires Cisco IOS-XE Release 3.1.0SG and later releases.
- WS-C4507R-E, WS-C4510R-E, WS-C4507R+E, and WS-C4510R+E are the only Catalyst 4500 series switches that support Supervisor Engine 7-E, Supervisor Engine 7L-E, and Supervisor Engine 8-E redundancy.
- In RPR or SSO mode,
 - with Supervisor Engine 7-E and 7L-E, only the first two uplinks on each supervisor engine are available. The second two uplinks are unavailable.
 - with Supervisor Engine 8-E, only the first four uplinks on each supervisor engine are available provided a 47xx line card is inserted on slot 10. The second set of four uplinks are unavailable. Only the first two uplinks on each supervisor are available unless the requirement is met (i.e., 47xx linecard in 4510 chassis).
- SSO requires both supervisor engines in the chassis to have the same components (model and memory), and to use the same Cisco IOS XE software image.
- The active and standby supervisor engines in the chassis must be in slots 3 and 4 for 7-slot chassis and slot 5 and 6 for 10-slot chassis.
- Each supervisor engine in the chassis must have its own flash device and console port connections to operate the switch on its own.
- Each supervisor engine must have a unique console connection. Do not connect a Y cable to the console ports.
- Supervisor engine redundancy does not provide supervisor engine load balancing.
- The Cisco Express Forwarding (CEF) table is cleared on a switchover. As a result, routed traffic is interrupted until route tables reconverge. This reconvergence time is minimal because the SSO feature reduces the supervisor engine redundancy switchover time from 30+ seconds to subsecond, so Layer 3 also has a faster failover time if the switch is configured for SSO.

- Static IP routes are maintained across a switchover because they are configured from entries in the configuration file.
- Information about Layer 3 dynamic states that is maintained on the active supervisor engine is not synchronized to the standby supervisor engine and is lost on switchover.
- If configuration changes on a redundant switch are made through SNMP set operations, the changes are not synchronized to the standby supervisor engine even in SSO mode. You might experience unexpected behavior.
- After you configure the switch through SNMP in SSO mode, copy the running-config file to the startup-config file on the active supervisor engine to trigger synchronization of the startup-config file to the standby supervisor engine. Then, reload the standby supervisor engine so that the new configuration is applied on the standby supervisor engine.
- You cannot perform configuration changes during the startup (bulk) synchronization. If you attempt to make configuration changes during this process, the following message is generated:

```
Config mode locked out till standby initializes
```
- If configuration changes occur at the same time as a supervisor engine switchover, these configuration changes are lost.
- If you remove a line card from a redundant switch and initiate an SSO switchover, then reinsert the line card, and all interfaces are shutdown. The rest of the original line card configuration is preserved.

This situation only occurs if a switch had reached SSO before you removed the line card.

Configuring Supervisor Engine Redundancy

These sections describe how to configure supervisor engine redundancy:

- [Configuring Redundancy, page 12-7](#)
- [Virtual Console for Standby Supervisor Engine, page 12-9](#)
- [Synchronizing the Supervisor Engine Configurations, page 12-10](#)

Configuring Redundancy



Note

IOS XE software can be booted at three different levels (Enterprise Services, IP Base, and LAN Base), based on the licenses available on the supervisor engine.

To configure redundancy, perform this task:

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

	Command	Purpose
Step 3	Switch# show running-config	Verifies that SSO or RPR is configured.
Step 4	Switch# show redundancy [clients counters history states]	Displays the redundancy information (counter, state, and so on) for the active and standby supervisor engines.

This example shows how to configure the system for SSO and display the redundancy facility information:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# redundancy
Switch(config-red)# mode sso
Switch(config-red)# end

Switch# show redundancy
Redundant System Information :
-----
      Available system uptime = 10 minutes Switchovers system experienced = 0
      Standby failures = 1
      Last switchover reason = none

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

Current Processor Information :
-----
      Active Location = slot 3
      Current Software state = ACTIVE
      Uptime in current state = 9 minutes
      Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
      (cat4500e-UNIVERSALK9-M), Version 15.0(100)XO(1.42), INTERIM SOFTWARE Copyright (c)
      1986-2010 by Cisco Systems, Inc.
      Compiled Sun 01-Aug-10 04:12 by gsbuprod
      Configuration register = 0x920

Peer Processor Information :
-----
      Standby Location = slot 4
      Current Software state = STANDBY HOT
      Uptime in current state = 0 minute
      Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
      (cat4500e-UNIVERSALK9-M), Version 15.0(100)XO(1.42), INTERIM SOFTWARE Copyright (c)
      1986-2010 by Cisco Systems, Inc.
      Compiled Sun 01-Aug-10 04:12 by gsbuprod
      Configuration register = 0x920
```

This example shows how to display redundancy facility state information:

```
Switch# show redundancy states
      my state = 13 -ACTIVE
      peer state = 8 -STANDBY HOT
      Mode = Duplex
      Unit = Primary
      Unit ID = 3

      Redundancy Mode (Operational) = Stateful Switchover
```



```

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

Communications = Up

client count = 64
client_notification_TMR = 240000 milliseconds
keep_alive TMR = 9000 milliseconds
keep_alive count = 1
keep_alive threshold = 18
RF debug mask = 0

```

This example shows how to change the system configuration from RPR to SSO mode:

```

Switch(config)# redundancy
Switch(config-red)# mode
Switch(config-red)# mode sso
Changing to sso mode will reset the standby. Do you want to continue?[confirm]
Switch(config-red)# end
Switch#
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-COMMUNICATION: Communication with the peer Supervisor
has been lost
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-SIMPLEX_MODE: The peer Supervisor has been lost

```

This example shows how to change the system configuration from SSO to RPR mode:

```

Switch(config)# redundancy
Switch(config-red)# mode rpr
Changing to rpr mode will reset the standby. Do you want to continue?[confirm]
Switch(config-red)# end
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-COMMUNICATION: Communication with the peer Supervisor
has been lost
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-SIMPLEX_MODE: The peer Supervisor has been lost

```

Virtual Console for Standby Supervisor Engine

Catalyst 4500 series switches can be configured with 2 supervisor engines to provide redundancy. When the switch is powered, one of the supervisor engines becomes active and remains active until a switchover occurs. The other supervisor engine remains in standby mode.

Each supervisor engine has its own console port. Access to the standby supervisor engine is possible only through the console port of the standby supervisor engine. Therefore, you must connect to the standby console to access, monitor or debug the standby supervisor.

Virtual Console for Standby Supervisor Engine enables you to access the standby console from the active supervisor engine without requiring a physical connection to the standby console. It uses IPC over EOBC to communicate with the standby supervisor engine and thus emulate the standby console on the active supervisor engine. Only one standby console session is active at any time.

The Virtual Console for Standby Supervisor Engine allows users who are logged onto the active supervisor engine to remotely execute **show** commands on the standby supervisor engine and view the results on the active supervisor engine. Virtual Console is available only from the active supervisor engine.

You can access the standby virtual console from the active supervisor engine with the **attach module**, **session module**, or **remote login** commands on the active supervisor engine. You must be in privilege EXEC mode (level 15) to run these commands to access the standby console.

Once you enter the standby virtual console, the terminal prompt automatically changes to *hostname-standby-console* where *hostname* is the configured name of the switch. The prompt is restored to the original setting when you exit the virtual console.

You exit the virtual console with the **exit** or **quit** commands. When the inactivity period of the terminal on the active supervisor engine where you logged in exceeds the configured idle time, you are automatically logged out of the terminal on the active supervisor engine. In this instance, the virtual console session is also terminated. Virtual console session is also automatically terminated when the standby is rebooted. After the standby boots up, you need to create another virtual console session.

To log in to the standby supervisor engine using a virtual console, do the following:

```
Switch# session module 4
Connecting to standby virtual console
Type "exit" or "quit" to end this session

Switch-standby-console# exit
Switch#
```

If the standby console is not enabled, the following message appears:

```
Switch-standby-console#
Standby console disabled.
Valid commands are: exit, logout
```



Note

The standby virtual console provides the standard features that are available from the supervisor console such as command history, command completion, command help and partial command keywords.

The following limitations apply to the standby virtual console:

- All commands on the virtual console run to completion. It does not provide the auto-more feature; it behaves as if the **terminal length 0** command has been executed. It is also non-interactive. Therefore, a running command cannot be interrupted or aborted by any key sequence on the active supervisor engine. If a command produces considerable output, the virtual console displays it on the supervisor engine screen.
- The virtual console is non-interactive. Because the virtual console does not detect the interactive nature of a command, any command that requires user interaction causes the virtual console to wait until the RPC timer aborts the command.

The virtual console timer is set to 60 seconds. The virtual console returns to its prompt after 60 seconds. During this time, you cannot abort the command from the key board. You must wait for the timer to expire before you continue.
- You cannot use virtual console to view debug and syslog messages that are being displayed on the standby supervisor engine. The virtual console only displays the output of commands that are executed from the virtual console. Other information that is displayed on the real standby console does not appear on the virtual console.

Synchronizing the Supervisor Engine Configurations

To manually synchronize the configurations used by the two supervisor engines, perform this task on the active supervisor engine:

	Command	Purpose
Step 1	Switch(config)# redundancy	Enters redundancy configuration mode.
Step 2	Switch(config-red)# main-cpu	Enters main-cpu configuration submenu.
Step 3	Switch(config-r-mc)# auto-sync { startup-config config-register bootvar standard }	Synchronizes the configuration elements.
Step 4	Switch(config-r-mc)# end	Returns to privileged EXEC mode.
Step 5	Switch# copy running-config startup-config	Synchronizes the running configuration in dynamic random-access memory (DRAM) to the startup configuration file in NVRAM. Note This step is not required to synchronize the running configuration file in (DRAM).

**Note**

Configuration changes made to the active supervisor engine through SNMP are not synchronized to the redundant supervisor engine. For information on how to handle this situation, see the “[Supervisor Engine Redundancy Guidelines and Restrictions](#)” section on page 12-5.

**Note**

The **auto-sync** command controls the synchronization of the config-reg, bootvar, and startup/private configuration files only. The calendar and VLAN database files are always synchronized when they change. In SSO mode, the running-config is always synchronized.

This example shows how to reenabling the default automatic synchronization feature using the **auto-sync standard** command to synchronize the startup-config and config-register configuration of the active supervisor engine with the standby supervisor engine. Updates for the boot variables are automatic and cannot be disabled.

```
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-sync standard
Switch(config-r-mc)# end
Switch# copy running-config startup-config
```

**Note**

To manually synchronize individual elements of the standard auto-sync configuration, disable the default automatic synchronization feature.

**Note**

When you configure the auto-sync standard, the individual sync options such as no auto-sync startup-config are ignored.

This example shows how to disable default automatic synchronization and allow only automatic synchronization of the config-registers of the active supervisor engine to the standby supervisor engine, while disallowing synchronization of the startup configuration:

```
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# no auto-sync standard
Switch(config-r-mc)# auto-sync config-register
Switch(config-r-mc)# end
```

Performing a Manual Switchover

This section describes how to perform a manual switchover (from the active supervisor engine to the standby supervisor engine) for test purposes. We recommend that you perform a manual switchover prior to deploying SSO in your production environment.



Note

This discussion assumes that SSO has been configured as the redundant mode.

To perform a manual switchover, perform this task on the active supervisor engine:

	Command	Purpose
Step 1	Switch# <code>show redundancy</code>	Verifies that the peer state is in the standby hot state. See the example of the <code>show redundancy states</code> command on page 6-10.
Step 2	Switch# <code>redundancy force-switchover</code>	Launches switchover from the active supervisor engine to the standby supervisor engine.

Be aware of these usage guidelines:

- force a switchover, the redundant supervisor engine must be in a standby hot (SSO) or standby cold (RPR) state. You can verify the state with the `show redundancy` command. If the state is not standby hot or standby cold, the `redundancy force-switchover` command will not execute.
- Use the `redundancy force-switchover` command, rather than the `reload` command, to initiate a switchover. The `redundancy force-switchover` command will first check that the redundant supervisor engine is in the correct state. If you issue the `reload` command and the status is not standby hot or standby cold, the `reload` command will reset the current supervisor engine and the peer supervisor may not be able to take over because it was not in a terminal state (standby hot or cold).

After a normal switchover, you might want to make the supervisor engine in a lower slot number of the chassis the active supervisor engine. Use the `show module` command to see which slot contains the active supervisor engine, and force another switchover if necessary.

Performing a Software Upgrade

This is useful only if IOS -XE software is running in LAN Base mode. For Enterprise Services or IP Base mode, use ISSU to upgrade software for both RPR and SSO redundant mode.

The software upgrade procedure supported by supervisor engine redundancy allows you to reload the Cisco IOS software image on the redundant supervisor engine, and once complete, reload the active supervisor engine once.

To perform a software upgrade, perform this task:

	Command	Purpose
Step 1	Switch# copy <i>source_device:source_filename</i> <i>slot0:target_filename</i> Or: Switch# copy <i>source_device:source_filename</i> bootflash: <i>target_filename</i>	Copies the new Cisco IOS-XE software image to bootflash on the supervisor engine.
Step 2	Switch# copy <i>source_device:source_filename</i> slaveslot0: <i>target_filename</i> Or: Switch# copy <i>source_device:source_filename</i> slavebootflash: <i>target_filename</i>	Copies the new image to a device on the standby supervisor engine (such as slavebootflash and slaveslot0).
Step 3	Switch# config terminal Switch(config)# config-register 0x2 Switch(config)# boot system flash <i>device:file_name</i>	Configures the supervisor engines to boot the new image. If your system was configured to autoboot an earlier image, issue the following command string to boot the new image instead: no boot system flash <i>device:old_file_name</i>
Step 4	Switch(config)# redundancy	Enters redundancy configuration mode.
Step 5	Switch(config-red)# main-cpu	Enters main-cpu configuration submode.
Step 6	Switch(config-r-mc)# auto-syn standard	Synchronizes the configuration elements.
Step 7	Switch(config-r-mc)# end	Returns to privileged EXEC mode.
Step 8	Switch# copy running-config start-config	Saves the configuration.
Step 9	Switch# redundancy reload peer	Reloads the standby supervisor engine and brings it back online (using the new release of the Cisco IOS-XE software).
Step 10	Switch# redundancy force-switchover	Conducts a manual switchover to the standby supervisor engine. The standby supervisor engine becomes the new active supervisor engine using the new Cisco IOS-XE software image. The old active supervisor engine reboots with the new image and becomes the standby supervisor engine.

This example shows how to perform a software upgrade:

```
Switch# config terminal
Switch(config)# config-register 0x2
Switch(config)# boot system flash
bootflash0:cat4500e-universalk9.SSA.03.01.00.150.1.XO.bin
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-syn standard
Switch(config-r-mc)# end
Switch# copy running-config start-config
Switch# redundancy reload peer
Switch# redundancy force-switchover
Switch#
```

This example illustrates how to verify that the running configuration on the active supervisor engine has successfully synchronized with the redundant supervisor engine:

```
Switch# config terminal
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-sync standard
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The bootvar has been successfully synchronized to the
standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The config-reg has been successfully synchronized to
the standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The startup-config has been successfully synchronized
to the standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The private-config has been successfully synchronized
to the standby supervisor
```

The example above shows that the boot variable, the config-register, and the startup configuration from the active supervisor engine have successfully synchronized to the redundant supervisor engine.

Manipulating Bootflash on the Standby Supervisor Engine



Note The console port on the standby supervisor engine is not available.

To manipulate the standby supervisor engine bootflash, perform one or more of the following tasks:

Command	Purpose
Switch# <code>dir slaveslot0:target_filename</code> or Switch# <code>dir slavebootflash:target_filename</code>	Lists the contents of the slot0: device on the standby supervisor engine. Lists the contents of the bootflash: device on the standby supervisor engine.
Switch# <code>delete slaveslot0:target_filename</code> or Switch# <code>delete slavebootflash:target_filename</code>	Deletes specific files from the slot0: device on the standby supervisor engine. Deletes specific files from the bootflash: device on the standby supervisor engine.
Switch# <code>squeeze slaveslot0:target_filename</code> or Switch# <code>squeeze slavebootflash:target_filename</code>	Squeezes the slot0: device on the standby supervisor engine. Squeezes the bootflash: device on the standby supervisor engine.
Switch# <code>format slaveslot0:target_filename</code> or Switch# <code>format slavebootflash:target_filename</code>	Formats the slot0: device on the standby supervisor engine. Formats the bootflash: device on the standby supervisor engine.
Switch# <code>copy source_device:source_filename slaveslot0:target_filename</code> or Switch# <code>copy source_device:source_filename slavebootflash:target_filename</code>	Copies a file from the active supervisor engine to the slot0: device on the standby supervisor engine. Copies a file to the bootflash: device on a standby supervisor engine.
	Note Source could be the active supervisor engine or a TFTP server.

■ Manipulating Bootflash on the Standby Supervisor Engine