



## Configuring Redundant Supervisor Engines

---

This chapter describes how to use and configure redundant supervisor engines on the Catalyst 6000 family switches.



**Note**

---

For more information about redundant operation with the Multilayer Switch Feature Card (MSFC), refer to the *Catalyst 6000 Family Multilayer Switch Feature Card and Policy Feature Card Configuration Guide*.

---



**Note**

---

For more information about installing redundant Catalyst 6000 family supervisor engines, refer to the *Catalyst 6000 Family Module Installation Guide*.

---



**Note**

---

For syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6000 Family Command Reference* publication.

---

This chapter consists of these sections:

- Understanding How Supervisor Engine Redundancy Works, page 16-2
- Synchronization Process Initiation, page 16-3
- Redundant Supervisor Engine Configuration Guidelines and Restrictions, page 16-4
- Verifying Standby Supervisor Engine Status, page 16-5
- Forcing a Switchover to the Standby Supervisor Engine, page 16-6
- High Availability Feature, page 16-7
- Supervisor Engine Synchronization Examples, page 16-14

# Understanding How Supervisor Engine Redundancy Works

**Note**

---

Redundant supervisor engines must be of the same type with the same model feature card.

---

When you install two supervisor engines, the first supervisor engine to come online becomes the active module; the second supervisor engine goes into standby mode. All administrative and network management functions, such as SNMP, command-line interface (CLI) console, Telnet, Spanning Tree Protocol (STP), Cisco Discovery Protocol (CDP), and VLAN Trunk Protocol (VTP) are processed on the active supervisor engine.

The console port on the standby supervisor engine is inactive and the module status for the standby supervisor engine shows as “standby.” However, status for the uplink ports on the standby supervisor engine is shown normally.

You must install redundant supervisor engines in slots 1 and 2 of the chassis. Redundant supervisor engines are hot swappable. The system continues to operate with the same configuration after switching over to the redundant supervisor engine. For more information, refer to the *Catalyst 6000 Family Module Installation Guide*.

**Note**

---

The switchover time from active to standby supervisor engine does not include spanning-tree convergence time.

---

At power-up, both supervisor engines run initial module-level diagnostics. Assuming both supervisor engines pass this level of diagnostics, the two supervisor engines communicate over the backplane, allowing them to cooperate during switching-bus diagnostics. The supervisor engine in slot 1 becomes active, and the supervisor engine in slot 2 enters standby mode. At this point, if the software versions of the two supervisor engines are different, or if the NVRAM configuration of the two supervisor engines is different, the active supervisor engine automatically downloads its software image and configuration to the standby supervisor engine.

If the background diagnostics on the active supervisor engine detect a major problem or an exception occurs, the active supervisor engine resets. The standby supervisor engine detects that the active supervisor engine is no longer running and becomes active. The standby supervisor engine can detect if the active supervisor engine is not functioning and can force a reset, if necessary. If the reset supervisor engine comes online again, it enters standby mode.

If you hot-insert a second supervisor engine, the second module communicates with the active supervisor engine after completing its initial module-level diagnostics. Because the active supervisor engine is already switching traffic on the backplane, no switching-bus diagnostics are run for the second supervisor engine because running diagnostics can disrupt normal traffic. The second supervisor engine immediately enters standby mode. The active supervisor engine downloads the software image and configuration to the standby supervisor engine, if necessary.

The supervisor engines use two Flash images: the *boot image* and the *runtime image*. The boot image filename is specified in the BOOT environment variable, which is stored in NVRAM. The runtime image is the boot image that the ROM monitor uses to boot the supervisor engine. After the system boots, the runtime image resides in dynamic RAM (DRAM).

When you power up or reset a switch with redundant supervisor engines, synchronization occurs to ensure that the runtime and boot images on the standby supervisor engine are the same as the images on the active supervisor engine.

The supervisor engines can have different runtime and boot images. If the boot image and the runtime image are the same, and you change the BOOT environment variable or overwrite or destroy the current boot image on the Flash device that was used to boot the system, the runtime and boot images will differ. Whenever you reconfigure the boot image, the active supervisor engine synchronizes its current boot image with the standby supervisor engine.

The boot image is read directly into the Flash file system. You can perform operations (such as **copy**, **delete**, **undelete**, and so on) on files stored on Flash memory devices, and you can store the boot image of the active supervisor engine in the standby supervisor engine bootflash. For more information about using the Flash file system, see Chapter 18, “Working With the Flash File System.”

The supervisor engine has a Flash PC card (PCMCIA) slot (**slot0:**) in addition to the onboard Flash memory; this slot can hold a Flash PC card that can store additional boot images.

**Note**

---

Throughout this publication, the term *Flash PC card* is used in place of the term *PCMCIA card*.

---

Since you can store multiple boot images, you must specify the name of the boot file image and the location of the image file in the Flash file system in order to boot and synchronize properly. For information about how to specify the name and location of the boot image, see Chapter 17, “Modifying the Switch Boot Configuration.”

In the synchronization process, the active supervisor engine checks the standby supervisor engine runtime image to make sure it matches its own runtime image. The active supervisor engine checks three conditions:

- If it needs to copy its boot image to the standby supervisor engine
- If the standby supervisor engine bootstring needs to be changed
- If the standby supervisor engine needs to be reset

The following section describes the conditions that can initiate Flash synchronization. For examples of how the system synchronizes the supervisor engine Flash images with various configurations, see the “Supervisor Engine Synchronization Examples” section on page 16-14.

## Synchronization Process Initiation

These conditions initiate the synchronization of the runtime and boot images on the active and standby supervisor engines:

- Timestamp mismatch between the runtime images on the active and standby supervisor engines—The active supervisor engine synchronizes its runtime image with the standby supervisor engine if the timestamps of their respective runtime images differ when the system is booted or reset.
- Timestamp mismatch between the boot images on the active and standby supervisor engines—The active supervisor engine synchronizes its boot image with the standby supervisor engine if the timestamps of their respective boot images differ when the system is booted or reset, or if you change the BOOT environment variable.
- Current boot image overwritten—If you overwrite the current boot image stored on one of the Flash devices, the file system management module detects this event and initiates synchronization. The active supervisor engine copies its new boot image to the standby supervisor engine.

- **BOOT environment variables changed**—If you change the BOOT environment variables to specify a different default boot image, the active supervisor engine initiates boot-image synchronization. The NVRAM configuration module detects this event and calls the Flash synchronization function with the next probable boot filename by looking at the boot configuration parameter.
- **Flash PC cards with same boot-image filename**—If you change the Flash device on either the active or standby supervisor engine and the new Flash device contains a boot image that has the same name (but a different timestamp) as the boot image from the previous Flash device, the Flash file management module initiates synchronization.
- **Current runtime image deleted**—If you delete the current runtime image from the Flash device, the Flash file management module prompts you to verify that you want to delete the current runtime image. If you confirm the deletion, the Flash file management module initiates Flash synchronization and informs the NVRAM configuration module of the change. The NVRAM configuration module examines the BOOT environment variable to determine the next probable image to boot and calls the Flash synchronization function using the new image name.

## Redundant Supervisor Engine Configuration Guidelines and Restrictions

The following conditions and events can cause the synchronization of images between redundant supervisor engines to fail or to produce unexpected results:

- **Downloading a new image to the active supervisor engine**

When you download a new image to the active supervisor engine, it is copied to the file system (in boot Flash or on a Flash PC card in the Flash PC card slot). Since you may or may not have configured this image as the boot image, the newly downloaded image is not copied to the standby supervisor engine automatically.

To initiate the synchronization function between the active and standby supervisor engines, you must configure this newly downloaded image as the boot image on the active supervisor engine. Synchronization occurs when you change the boot variable. To run the new image, you must reset the system.

- **Unable to find the current runtime image**

If the active supervisor engine is unable to find the current runtime image on any of the Flash devices, it signals an error condition. In this case, if the standby supervisor engine is inserted or reset, Flash synchronization does not occur. In addition, the STATUS LED on the standby supervisor engine turns red and the system generates a syslog error message.

- **Active supervisor engine in slot 2**

When the active supervisor engine is in slot 2, the standby supervisor engine is in slot 1. If you change the configuration to specify a new boot image and then reset the system, the supervisor engine in slot 1 becomes the active supervisor engine and loads its default boot image, canceling the configuration changes you have just made. To avoid this problem, the switch prompts you for Flash synchronization as soon as you change the boot file configuration.

# Verifying Standby Supervisor Engine Status

You can verify the status of the standby supervisor engine using a number of CLI commands.



**Note**

The **show module** output provides information about installed daughter cards. The **show test** command provides information about onboard application-specific integrated circuits (ASICs).

To verify the status of the standby supervisor engine, perform one or more of these tasks:

| Task  | Command   |
|---|---|
| <ul style="list-style-type: none"> <li>Show the status of the standby supervisor engine.</li> </ul>               | <b>show module</b> [ <i>mod_num</i> ]                   |
| <ul style="list-style-type: none"> <li>Show the state of the standby supervisor engine uplink ports.</li> </ul>   | <b>show port</b> [ <i>mod_num</i> [/ <i>port_num</i> ]] |
| <ul style="list-style-type: none"> <li>Show diagnostic test results for the standby supervisor engine.</li> </ul> | <b>show test</b> [ <i>mod_num</i> ]                     |

This example shows how to check the status of the standby supervisor engine using the **show module** and **show test** commands:

```

Console> (enable) show module 2
Mod Slot Ports Module-Type           Model              Status
-----
2   2   2   1000BaseX Supervisor      WS-X6K-SUP1-2GE   ok

Mod Module-Name          Serial-Num
-----
2                          SAD02330231

Mod MAC-Address(es)      Hw    Fw    Sw
-----
2   00-e0-14-0e-f5-6c to 00-e0-14-0e-f5-6d 0.404  4.2(2038)  4.2(0.24)VAI50
   00-e0-14-0e-f5-6e to 00-e0-14-0e-f5-6f
   00-10-7b-bb-2b-00 to 00-10-7b-bb-2e-ff

Mod Sub-Type            Sub-Model          Sub-Serial  Sub-Hw
-----
2   L2 Switching Engine WS-F6020           SAD02350211 0.101
Console> (enable)

Console> (enable) show test 2
Module 2 : 2-port 1000BaseX Supervisor
Network Management Processor (NMP) Status: (. = Pass, F = Fail, U = Unknown)
ROM: .   Flash-EEPROM: .   Ser-EEPROM: .   NVRAM: .   EOBC Comm: .

Line Card Status for Module 1 : PASS

Port Status :
Ports 1 2
-----
. .

```



```

System Power On Diagnostics
NVRAM Size .. .....512KB
ID Prom Test .....Passed
DPRAM Size .....16KB
DPRAM Data 0x55 Test .....Passed
DPRAM Data 0xaa Test .....Passed
DPRAM Address Test .....Passed
Clearing DPRAM .....Done
System DRAM Memory Size .....32MB
DRAM Data 0x55 Test .....Passed
DRAM Data 0xaa Test .....Passed
DRAM Address Test .....Passed
Clearing DRAM .....Done
EARLII .....Present
EARLII RAM Test .....Passed
EARL Serial Prom Test .....Passed
Level2 Cache .....Present
Level2 Cache test.....Passed

Boot image: bootflash:cat6000-sup.5-4-1a.bin
Downloading epld sram device please wait ...
Programming successful for Altera 10K50 SRAM EPLD
This module is now in standby mode.
Console is disabled for standby supervisor

```

This example shows the console output on the standby supervisor engine when you force a switchover from the active to the standby supervisor engine:

```

Cisco Systems Console

Enter password:
12/07/1998,17:04:43:MLS-5:Multilayer switching is enabled
12/07/1998,17:04:43:MLS-5:Netflow Data Export disabled
12/07/1998,17:04:44:SYS-5:Module 2 is online
12/07/1998,17:04:45:SYS-5:Module 5 is online
12/07/1998,17:04:45:SYS-5:Module 7 is online
12/07/1998,17:04:45:SYS-5:Module 3 is online
12/07/1998,17:04:52:MLS-5:Route Processor 172.20.52.6 added
12/07/1998,17:05:10:SYS-5:Module 8 is online
12/07/1998,17:05:14:SYS-5:Module 9 is online
12/07/1998,17:05:22:SYS-5:Module 4 is online
12/07/1998,17:06:13:SYS-5:Module 1 is in standby mode
Supervisor image synchronization process will start in 10 seconds
12/07/1998,17:06:37:SYS-5:Ports on standby supervisor(Module 1) are UP
12/07/1998,17:06:41:SYS-5:Active supervisor is synchronizing the NMP image.
12/07/1998,17:06:44:SYS-5:The active supervisor has synchronized the NMP image.

Console>

```

## High Availability Feature

This section describes the high availability feature that is used to minimize the switchover time from the active supervisor engine to the standby supervisor engine if the active supervisor engine fails. The high availability feature ensures that there will be minimal disturbance to the steady-state operation of the network when the standby supervisor engine takes control of the system.

Prior to the high availability feature, the *fast switchover* feature was available to ensure that a switchover to the standby supervisor engine happened quickly. However, with fast switchover, all the switch features needed to be reinitialized and restarted when the standby supervisor engine assumed the active role because their state before the switchover was unknown.

The high availability feature removes this limitation; the active supervisor engine communicates with the standby, keeping feature protocol states synchronized. With the standby supervisor engine synchronized with the active supervisor engine, the standby can take over in the event of a failure and continue exactly where the failed supervisor engine left off.

The high availability feature also provides a *versioning* option that allows you to run different software images on the active and standby supervisor engines.

These features are discussed in detail in the following sections:

- High Availability Overview, page 16-8
- High-Availability Supported Features, page 16-9
- Versioning Overview, page 16-10
- CLI Commands, page 16-11
- Loading a Different (but Compatible) Image on the Standby Supervisor Engine, page 16-13

## High Availability Overview

For high availability, a system database is maintained on the active supervisor engine and updates are sent to the standby supervisor engine for any change of data in the system database. The active supervisor engine communicates and updates the standby supervisor engine when any state changes occur, ensuring that the standby supervisor engine knows the current protocol state of supported features. The standby supervisor engine knows the current protocol states for all modules, ports, and VLANs; the protocols can initialize with this state information and start running immediately.



### Note

See Table 16-1 on page 16-10 for a list of high availability-supported features.

The active supervisor engine controls the system bus (backplane), sends and receives packets to and from the network, and controls all modules. Protocols run on the active supervisor engine only.

The standby supervisor engine is isolated from the system bus and does not switch packets. But it *does* receive packets from the switching bus to learn and populate its Layer 2 forwarding table for Layer 2-switched flows. The standby supervisor engine also receives packets from the switching bus to learn and populate the Multilayer Switching (MLS) table for Layer 3-switched flows. The standby supervisor engine does not participate in forwarding any packets and does not communicate with any modules.



### Note

Routing table entries in the active MSFC are not preserved because the high availability feature is not run on the MSFC IOS. However, you can configure both MSFCs on the active and standby supervisor engines with the same configuration and use Hot Standby Router Protocol (HSRP) to preserve routing table entries across the active and standby MSFCs. For detailed information on configuring the MSFC, refer to the *Catalyst 6000 Family Multilayer Switch Feature Card and Policy Feature Card Configuration Guide* publication.

When you enable high availability, and the standby supervisor engine is running, image version compatibility is checked and if found compatible, the database synchronization is started. High availability compatible features continue from the saved states on the standby supervisor engine after a switchover.

When you disable high availability, the database synchronization is not done and all features must restart on the standby supervisor engine after a switchover.

If you change high availability from enabled to disabled, synchronization from the active supervisor engine is stopped and the standby supervisor engine discards all current synchronization data.

If you change high availability from disabled to enabled, synchronization from the active to standby supervisor engine is started (provided the standby supervisor engine is present and its image version is compatible).

NVRAM synchronization occurs irrespective of high availability being enabled or disabled (provided there are compatible NVRAM versions on the two supervisor engines).

**Note**

---

See the “Versioning Overview” section on page 16-10 for version compatibility details.

---

If the standby supervisor engine is not installed during system bootup, the active supervisor engine detects this and the database updates are not queued for synching. Similarly, when you reset or remove the standby supervisor engine, the synchronization updates are not queued and any pending updates in the synchronization queue are discarded. When you hot insert or restart a second supervisor engine that becomes the standby supervisor engine, the active supervisor engine downloads the entire system database to the standby supervisor engine. Only after this global synchronization is completed, does the active supervisor engine queue and synchronize the individual updates to the standby supervisor engine.

**Note**

---

When you hot insert or restart a second supervisor engine, it might take a few minutes for the global synchronization to complete.

---

## High-Availability Supported Features

**Note**

---

MLS flows are preserved from the active supervisor engine to the standby supervisor engine.

---

The Catalyst 6000 family switch features are classified into three categories with respect to high availability support (see Table 16-1):

- Supported features—High availability is fully supported; the feature’s database is synchronized from the active supervisor engine to the standby supervisor engine.
- Compatible features—High availability is not supported; the feature’s database is not synchronized from the active supervisor engine to the standby supervisor engine. However, **the feature can be enabled** (operational) along with the high availability feature.
- Incompatible features—High availability is not supported; the feature’s database is not synchronized from the active supervisor engine to the standby supervisor engine. Additionally, **the feature cannot be enabled** if high availability is enabled, and similarly, high availability cannot be enabled if the feature is enabled.

**Note**

---

Timers and statistics are not synchronized from the active to the standby supervisor engine.

---

**Table 16-1 High Availability Feature Support**

| Supported Features | Compatible Features | Incompatible Features |
|--------------------|---------------------|-----------------------|
| COPS-DS            | ASLB                | Dynamic VLAN          |
| COPS-PR            | CDP                 | GVRP                  |
| DTP                | GMRP                | Port security         |
| EtherChannel       | IGMP snooping       | Protocol filtering    |
| IOS ACLs           | RMON                |                       |
| MLS                | RSVP                |                       |
| PAgP               | SNMP                |                       |
| QoS                | Telnet sessions     |                       |
| SPAN               | UplinkFast          |                       |
| STP                | VTP pruning         |                       |
| Trunking           |                     |                       |
| UDLD               |                     |                       |
| VACLs              |                     |                       |
| VTP                |                     |                       |

## Versioning Overview

With high availability versioning enabled, you can have two different but compatible images on the active and standby supervisor engines. The active supervisor engine exchanges image version information with the standby supervisor engine and determines whether the images are compatible for enabling high availability. If the active and standby supervisor engines are not running compatible image versions, the high availability feature cannot be enabled.

Image versioning is supported in supervisor engine software releases 5.4(1)CSX and later. Image versioning is not supported with images prior to release 5.4(1)CSX. Therefore, with versioning enabled, the high availability feature is fully supported with the active and standby supervisor engines running different images as long as both images are release 5.4(1)CSX or later.



### Note

When you install two supervisor engines, the first supervisor engine to come online becomes the active module; the second supervisor engine goes into standby mode. In a system with two supervisor engines installed, at power up the supervisor engine in slot 1 becomes active, and the supervisor engine in slot 2 enters standby mode. At this point, if the software versions of the two supervisor engines are different, or if the NVRAM configuration of the two supervisor engines is different, **and versioning is not enabled**, the active supervisor engine automatically downloads its software image and configuration to the standby supervisor engine.

## CLI Commands

This section describes the CLI commands associated with high availability and versioning.

### Enabling or Disabling High Availability

High availability is disabled by default. To enable or disable high availability, perform these tasks in privileged mode:

| Task                                 | Command   |
|--------------------------------------|---|
| Enable or disable high availability. | <b>set system highavailability enable   disable</b> |

This example shows how to enable high availability:

```
Console> (enable) set system highavailability enable
System high availability enabled.
Console> (enable)
```

This example shows how to disable high availability:

```
Console> (enable) set system highavailability disable
System high availability disabled.
Console> (enable)
```

### Enabling or Disabling High Availability Versioning

High availability versioning is disabled by default. To enable or disable high availability versioning, perform these tasks in privileged mode:

| Task  | Command  |
|---|--|
| Enable or disable high availability versioning. | <b>set system highavailability versioning enable   disable</b> |

This example shows how to enable high availability versioning:

```
Console> (enable) set system highavailability versioning enable
Image versioning enabled.
Console> (enable)
```

This example shows how to disable high availability versioning:

```
Console> (enable) set system highavailability versioning disable
Image versioning disabled.
Console> (enable)
```

### Showing High Availability Settings and Operational Status

The **show system highavailability** command displays the following:

- High availability setting (enabled or disabled)
- Versioning setting (enabled or disabled)

- High availability operational status (based on whether the standby supervisor engine is present and operational). The operational status field displays one of the following:
  - OFF (high-availability-not-enabled): The high availability option in NVRAM is disabled.
  - OFF (standby-supervisor-not-present): The standby supervisor engine is not installed.
  - OFF (standby-supervisor-image-incompatible): The standby supervisor engine is running a different image than the active supervisor engine and it is not version compatible (the versioning option in NVRAM is enabled). No synchronization is done (even a configuration change in NVRAM on the active supervisor engine cannot be propagated to the standby because of the version incompatibility).
  - OFF (standby-supervisor-image-nvram-only-compat): The standby supervisor engine is running a different image than the active supervisor engine (versioning option in NVRAM is enabled) and the image is only NVRAM compatible (that is, a configuration change in NVRAM on the active supervisor engine will be propagated to the standby). However, high availability cannot be supported.
  - OFF (standby-supervisor-not-operational-yet): The standby supervisor engine is detected, but is not operational (not online yet).
  - OFF (high-availability-not-operational-yet): The standby supervisor engine is operational (online), but high availability is not operational yet (when the system is booted from reset, it takes a few minutes before high availability is operational).
  - ON: High availability is operational. The active supervisor engine's features have started queuing their state changes for synchronizing to the standby supervisor engine.

To show the high availability configuration and operational states, perform this task:

| Task   | Command                             |
|--|-------------------------------------|
| Show high availability configuration and operational states. | <b>show system highavailability</b> |

In this example, high availability and versioning are disabled:

```
Console> (enable) show system highavailability
Highavailability: disabled
Highavailability versioning: disabled
Highavailability Operational-status: OFF (high-availability-not-enabled)
Console> (enable)
```

In this example, high availability is enabled:

```
Console> (enable) set system highavailability enable
System high availability enabled.
Console> (enable)

Console> (enable) show system highavailability
Highavailability: enabled
Highavailability versioning: disabled
Highavailability Operational-status: ON
Console> (enable)
```

## Loading a Different (but Compatible) Image on the Standby Supervisor Engine

Use this procedure to load a new image on the standby supervisor engine that is different from the image on the active supervisor engine. From the active supervisor engine console port, perform these steps (active supervisor engine is in slot 1):

---

**Step 1** Enable high availability versioning.

```
Console> (enable) set system highavailability enable
System high availability enabled.
Console> (enable)
```

**Step 2** Download the new image to the active supervisor engine bootflash.

```
Console> (enable) copy tftp:image2.bin bootflash
IP address or name of remote host []? 172.20.52.3

8763532 bytes available on device bootflash, proceed (y/n) [n]? y

... display text truncated
Console> (enable)
```

**Step 3** Copy the new image to the standby supervisor engine bootflash.

```
Console> (enable) copy bootflash:image2.bin 2/bootflash:

5786532 bytes available on device bootflash, proceed (y/n) [n]? y

... display text truncated
Console> (enable)
```

**Step 4** Modify the BOOT environment variable so the standby supervisor engine boots the new image.

```
Console> (enable) set boot system flash bootflash:image2.bin prepend 2
BOOT variable = bootflash:image2.bin,1;slot0:image1.bin,1
Console> (enable)
```

**Step 5** To boot the new image, reset the standby supervisor engine.

```
Console> (enable) reset 2
This command will reset the system.
Do you want to continue (y/n) [n]? y

... display text truncated
Console> (enable)
```

---

# Supervisor Engine Synchronization Examples

The following examples show what happens when the synchronization function encounters certain conditions. These examples are not intended to cover every possible condition.



## Note

In the following examples, the number **1** following the filename in the bootstring (for example, **bootflash:f1,1**) indicates the number of Trivial File Transfer Protocol (TFTP) boot retries that are attempted. However, the supervisor engine does not support TFTP booting. The number is included in these examples to be consistent with Cisco IOS conventions.

## Synchronizing the Runtime Image with the Bootstring

This section contains four examples in which the active supervisor engine runtime image is synchronized with the standby supervisor engine.

### Example 1: Runtime image not synchronized

The configuration for example 1 is as follows:

- The active supervisor engine configuration is as follows (if the image in the standby supervisor engine is identical to the image in the active supervisor engine, the output is the same):
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1`
  - Bootflash: `f1`
- The timestamp for **f1** on the active supervisor engine is the same as **f1** on the standby supervisor engine.
- Expected result:
  - The active supervisor engine **f1** image is not copied to the standby supervisor engine.
  - The standby supervisor engine bootstring is not modified.
  - The standby supervisor engine is not reset.

### Example 2: File copied, bootstring changed, standby supervisor engine reset

The configuration for example 2 is as follows:

- Active supervisor engine configuration:
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1`
  - Bootflash: `f1`
- Standby supervisor engine configuration:
  - Runtime image: `bootflash:f2`
  - Boot string: `bootflash:f2,1`
  - Bootflash: `f2`

- The timestamp for **f1** on the active supervisor engine is not the same as **f2** on the standby supervisor engine.
- Expected result:
  - The active supervisor engine copies **f1** to the standby supervisor engine and renames the file **RTSYNC\_f1**.
  - The standby supervisor engine bootflash is modified to the following: **f2, RTSYNC\_f1**.
  - The standby supervisor engine bootstring is modified to the following: **bootflash:RTSYNC\_f1,1;f2,1;**
  - The standby supervisor engine is reset.

### Example 3: File not copied, bootstring changed, standby supervisor engine reset

The configuration for example 3 is as follows:

- Active supervisor engine configuration:
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1`
  - Bootflash: `f1`
- Standby supervisor engine configuration:
  - Runtime image: `bootflash:f2`
  - Boot string: `bootflash:f2,1`
  - Bootflash: `f1, f2`
- The timestamp for **f1** on the active supervisor engine is the same as **f1** on the standby supervisor engine but is not the same as **f2** on the standby supervisor engine.
- Expected result:
  - The active supervisor engine runtime image is synchronized to the standby supervisor engine.
  - The active supervisor engine **f1** image is not copied to the standby supervisor engine.
  - The standby supervisor engine boot string is modified to the following: **f1,1;f2,1;**
  - The standby supervisor engine is reset.

### Example 4: Oldest bootflash file deleted, bootflash squeezed

The configuration for example 4 is as follows:

- Active supervisor engine configuration:
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1`
  - Bootflash: `f1`
- Standby supervisor engine configuration:
  - Runtime image: `bootflash:f2`
  - Boot string: `bootflash:f2,1;`
  - Bootflash: `f2, f3, f4` (less than 1 MB left on device)

- The timestamp for **f1** on the active supervisor engine is not the same as **f2** on the standby supervisor engine. The **f2** timestamp is older than **f3**, and the **f3** timestamp is older than **f4**.
- Expected result:
  - The active supervisor engine runtime image is synchronized with the standby supervisor engine.
  - The active supervisor engine attempts to copy its **f1** image to the standby supervisor engine.
  - Since there is not enough space on the standby supervisor engine bootflash, the redundant synchronization function finds the oldest file, deletes it, and squeezes bootflash.
  - The active supervisor engine copies the **f1** image to the standby supervisor engine and renames it **RTSYNC\_f1**.
  - The standby supervisor engine bootflash is modified to the following: **f3, f4, RTSYNC\_f1**.
  - The standby supervisor engine boot string is modified to the following: **RTSYNC\_f1,1;f2,1;**
  - The standby supervisor engine is reset.

## Synchronizing the Boot Images on the Active and Standby Supervisor Engines

This section contains four examples in which the bootstrings on the active and standby supervisor engines are synchronized.

### Example 1: Unable to allocate the boot image

The configuration for this example is as follows:

- Active supervisor engine configuration:
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1;`
  - Bootflash: `f1`
- Standby supervisor engine configuration:
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1;`
  - Bootflash: `f1`
- The timestamp for **f1** on the active supervisor engine is the same as **f1** on the standby supervisor engine.
- The system attempts to modify the active supervisor engine bootstring to the following: **f2,1;**
- Expected result:
  - The active supervisor engine is unable to allocate **f2**, causing the synchronization to fail.
  - An error is recorded in `syslog`.
  - The active supervisor engine **f1** image is not copied to the standby supervisor engine.
  - The standby supervisor engine bootstring is not modified.
  - The standby supervisor engine is not reset.

## Example 2: File copied, bootflash modified, standby supervisor engine not reset

The configuration for this example is as follows:

- Active supervisor engine configuration:
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1;`
  - Bootflash: `f1, f2`
- Standby supervisor engine configuration:
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1;`
  - Bootflash:
- The timestamp for **f1** on the active supervisor engine is the same as **f1** on the standby supervisor engine.
- You modify the active supervisor engine bootstring to the following: **f2,1;**.
- Expected result:
  - The active supervisor engine copies its **f2** image to the standby supervisor engine and renames it **BTSYNC\_f2**.
  - The standby supervisor engine bootflash is modified to the following: **f1, BTSYNC\_f2**.
  - The standby supervisor engine bootstring is modified to the following: **bootflash:BTSYNC\_f2,1;f1,1;**
  - The standby supervisor engine is not reset.

## Example 3: File not copied, bootstring modified, standby supervisor engine not reset

The configuration for this example is as follows:

- Active supervisor engine configuration:
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1;`
  - Bootflash: `f1, f2`
- Standby supervisor engine configuration:
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1;`
  - Bootflash: `f1, f2`
- The timestamp for **f1** on the active supervisor engine is the same as **f1** on the standby supervisor engine; the timestamp for **f2** on the active supervisor engine is the same as **f2** on the standby supervisor engine.
- The active supervisor engine bootstring is modified to the following: **f2,1; f1,1;**.
- Expected result:
  - The active supervisor engine **f1** image is not copied to the standby supervisor engine.
  - The standby supervisor engine bootstring is modified to the following: **bootflash:f2,1;bootflash:f1,1;**
  - The standby supervisor engine is not reset.

## Example 4: File copied, oldest file deleted, bootflash squeezed, bootstring modified, standby supervisor engine not reset

The configuration for this example is as follows:

- Active supervisor engine configuration:
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1;`
  - Bootflash: `f1, f2`
- Standby supervisor engine configuration:
  - Runtime image: `bootflash:f1`
  - Boot string: `bootflash:f1,1;`
  - Bootflash: `f0, f1, f3` (less than 1 MB left on device)
- The timestamp for **f1** on the active supervisor engine is the same as **f1** on the standby supervisor engine. The timestamp for **f0** is older than **f1**, and the timestamp for **f1** is older than **f3**.
- The active supervisor engine bootstring is modified to the following:  
**bootflash:f2,1;bootflash:f1,1;**
- Expected result:
  - The active supervisor engine attempts to copy its **f2** image to the standby supervisor engine.
  - Since there is not enough space available on the standby supervisor engine bootflash, the redundant synchronization function finds the oldest file (**f0**), deletes it, and squeezes bootflash.
  - The active supervisor engine copies its **f2** image to the standby supervisor engine and renames it **BTSYNC\_f2**.
  - The standby supervisor engine bootflash is modified to the following: **f1, f3, BTSYNC\_f2**.
  - The standby supervisor engine boot string is modified to the following:  
**bootflash:BTSYNC\_f2,1;bootflash:f1,1;**