

Cisco IOS Software In-Service Software Upgrade on the Cisco Catalyst 4500 Series

This document describes the In-Service Software Upgrade (ISSU) process and functionality implemented on the Cisco® Catalyst® 4500 Series. It examines step-by-step instructions for performing in-service software version changes or maintenance on the Cisco Catalyst 4500 redundant series. This document is not meant to be a complete introduction to high availability, but rather to describe the specifics as they apply to the Cisco Catalyst 4500 Series and ISSU.

What Is ISSU?

ISSU is available on the Cisco Catalyst 4500 Series and allows customers to virtually eliminate planned outages for full feature software upgrades. It provides the means to upgrade or, if needed, downgrade the Cisco IOS® Software in a redundant Cisco Catalyst 4500 supervisor system *without* incurring a service outage. ISSU adds additional functionality to the Cisco Catalyst 4500 high-availability capabilities already provided by stateful switchover (SSO) and nonstop forwarding (NSF). Since the underlying technology supporting ISSU is based on the SSO architecture, the downtime associated during a switchover is less than 200 ms. ISSU/SSO and NSF are explained in detail in the following sections of this paper.

ISSU is a user-initiated and user-controlled process through a set of executive-level CLI commands issued in a specific order to upgrade or downgrade a Cisco IOS Software image running on a Cisco Catalyst 4500 dual-supervisor configuration and differs from “hitless” software upgrades in that it provides the ability to do a hitless “full feature” upgrade rather than just a system patch.

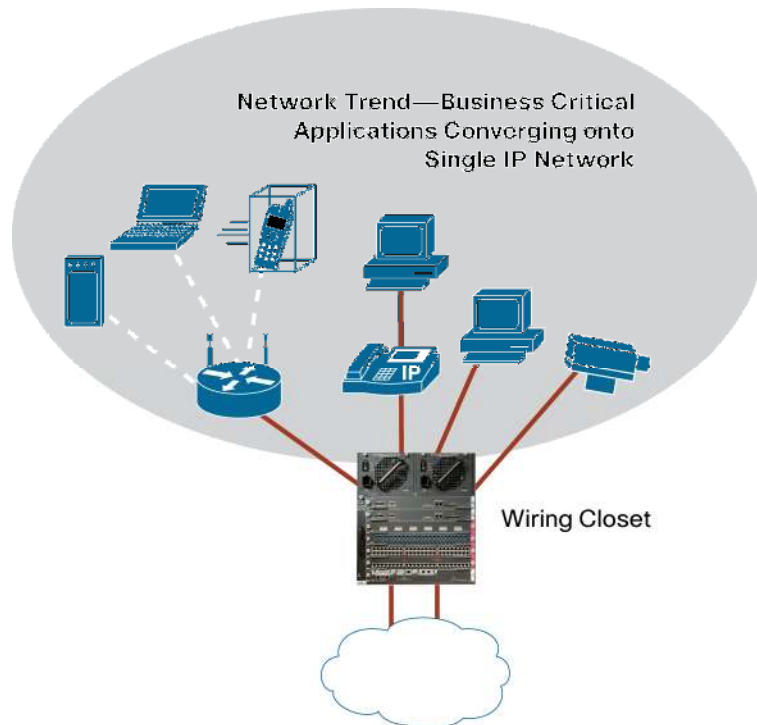
ISSU was introduced with Cisco IOS Software Release 12.2(31)SGA.

The benefits of ISSU are summarized in Table 1.

Table 1. ISSU Benefits

Operational Advantages	Business Advantages
Eliminates network downtime for Cisco IOS Software image upgrades	Business continuity: “always on” network Increased user productivity by eliminating scheduled outages
Faster implementation of new features, hardware, and fixes	Accelerates deployment of new services and applications
Eliminates “late night” maintenance windows for software upgrades	Mitigates security risks with timely fixes transparent to end users
Mitigates upgrade risk with rapid rollback feature	Provides added insurance for a successful upgrade
Reduces operational costs while delivering higher SLA	Reduces operational costs No more “network will be down for software upgrade” emails

A current networking trend, which is accelerating, is to converge business-critical applications onto a single IP network for unified communications, as shown in Figure 1.

Figure 1. Networking Trend for Business-Critical Applications

This single point of connectivity requires the highest system availability. With the addition of ISSU, the Cisco Catalyst 4500 offers an extremely comprehensive high-availability feature set. These features help ensure that the Cisco Catalyst 4500 delivers the highest total system availability for deploying business critical and real-time applications over a converged IP network.

Cisco Catalyst 4500 High Availability

It is important to note the evolution of the Cisco Catalyst 4500 high-availability initiative. The next section discusses the functionality of each redundancy mode and how ISSU complements the proven NSF/SSO architecture. For complete details on how to configure NSF/SSO on the Cisco Catalyst 4500, visit

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/nsfwss0.htm.

Route Processor Redundancy

Before the SSO mode of redundancy was available, customers were given the option to configure a redundant Cisco Catalyst 4500 chassis in Route Processor Redundancy (RPR) mode. RPR was initially supported on the Cisco Catalyst 4500 with Cisco IOS Software Release 12.1(12c)EW. With RPR mode the following behavior occurs:

- The standby supervisor performs a partial boot and suspends at the Cisco IOS Software init process.
- All persistent-configurations (startup-config, private-config, vlan-database, boot variables, config-register, calendar, and clock) are synchronized from the active supervisor to the standby supervisor; however the running-config is not synchronized.
- When changes are made to the persistent configurations, the updated data is synchronized to the standby.
- The standby supervisor monitors the health of the active supervisor. If the active supervisor fails, the standby takes over and continues to initialize.
- The standby supervisor discovers the line cards in the chassis by reading the line card idproms and parses its startup-config.
- During a switchover, the line cards are reset; resulting in lost traffic. Approximate switchover time is ~ 60 seconds.

NSF/SSO

The RPR mode was adopted by customers and soon began to pave the way for the next level of supervisor redundancy. In September of 2004, the Cisco Catalyst 4500 introduced its capability to support the SSO mode of supervisor redundancy. SSO was initially supported on the Cisco Catalyst 4500 with Cisco IOS Software Release 12.2(20)EWA. With SSO, the functionality of the supervisors is the following:

- The standby supervisor performs a complete boot without suspending at the Cisco IOS Software init process.
- Each feature synchronizes its internal state between the active and standby supervisors.
- Changes made to the running configuration are synchronized to the standby.
- Physical links remain up, and L2 protocols are not reset, yielding minimal packet loss, less than 200 ms of traffic loss.

SSO is the component of the solution that synchronizes and saves state information between the active and standby supervisors such that Layer 2 connectivity protocols are maintained.

In order to preserve L3 protocol state information, NSF was developed. As the acronym states, NSF allows for continued forwarding of data packets along known routes while the routing protocol information is recovered and validated gracefully, avoiding unnecessary route flaps and network instability. The NSF capability was introduced on the Cisco Catalyst 4500 in Cisco IOS Software Release 12.2(31)SG. NSF capability requires a redundant system to run in SSO mode.

How Does ISSU Work?

ISSU should be thought of as a process or procedure allowing customers to upgrade or if needed downgrade a Cisco IOS Software image running on a Cisco Catalyst 4500 system configured for SSO/NSF, from a lower version to higher version or vice versa. This process moves a Cisco Catalyst 4500 from one version of SSO/NSF-capable Cisco IOS Software image to another version of SSO/NSF-capable Cisco IOS Software image with minimized downtime, degradation of service, or loss of packets.

In order to perform ISSU while the supervisor concurrently forwards packets, you must first have the following:

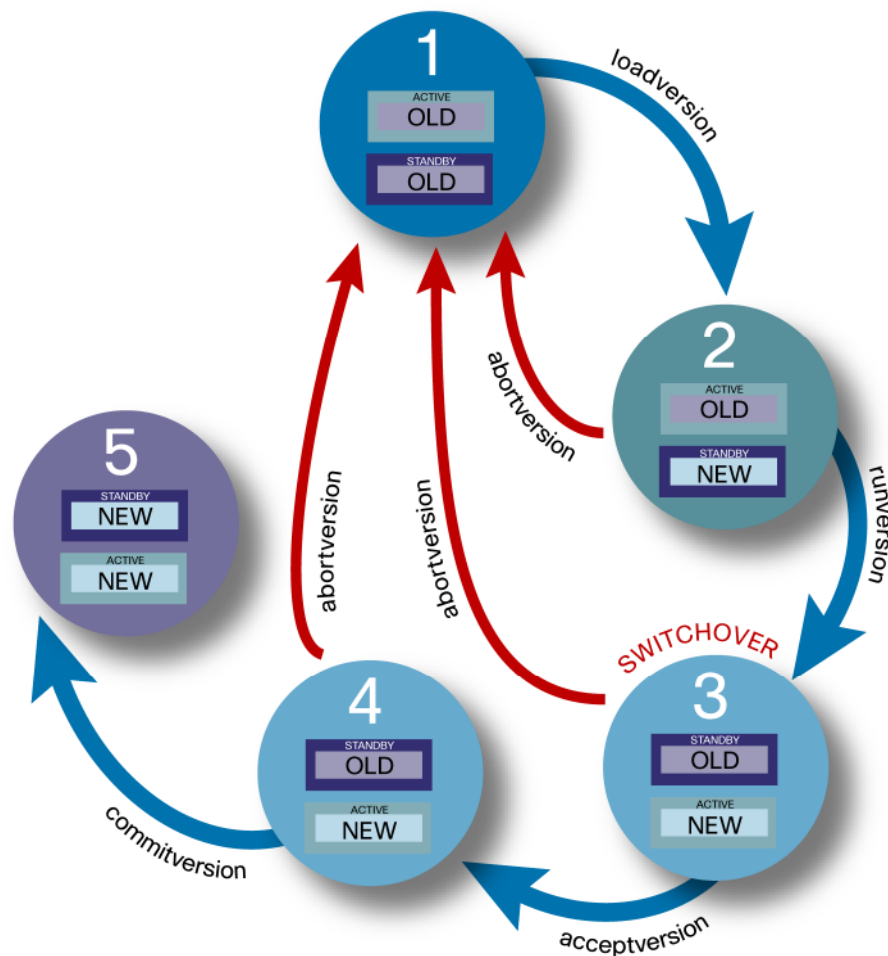
- A redundant Cisco Catalyst 4500 chassis (4507R or 4510R)
- Previous implementation of Cisco NSF/SSO

The ISSU itself is a user-initiated and user-controlled process through a set of exec-level CLI commands issued in a certain specific order.

In essence, the procedure (see Figure 2) can be simply described as follows:

1. Reset the standby supervisor with the new software.
2. Switch over to the standby supervisor with the new software; making it the active supervisor.
3. Reset the new standby supervisor (the original active supervisor) with the new software.

Figure 2. ISSU Process and Associated Commands



This process and the commands, along with their detailed behavior, are described in the following sections of this document.

These commands can also be referenced in the “Configuring the Cisco IOS Software In Service Software Upgrade Process” in the Cisco IOS Software Release 12.2(31)SGA Configuration Guide at

http://www.cisco.com/en/US/products/hw/switches/ps4324/products_configuration_guide_chapter09186a00806c3604.html.

ISSU Prerequisites

Before one can perform an ISSU, there are a few prerequisites one must verify for a successful ISSU. The following list explains what is initially required.

- Must be using a redundant Cisco Catalyst 4500 switch with symmetric hardware (that is, supervisors, memory, rommon, NFL daughter card, and so on).
- Both new and old Cisco IOS Software images must be preloaded to the file system on both supervisors.
- SSO must be configured and working properly.
- Config register must be configured to autoboot (that is, the value should have a “2” in the lowest byte).

```
45010R-203#sh bootvar | i register
Configuration register is 0x2102
Standby Configuration register is 0x2102
```

Several commands are available to verify if SSO is enabled:

```
4510R-203#sh module | b Redundancy
Mod  Redundancy role      Operating mode      Redundancy status
----+-----+-----+-----+-----
  1   Standby Supervisor  SSO                 Standby hot
  2   Active Supervisor   SSO                 Active
```

```
45010R-203#sh redundancy states
my state = 13 -ACTIVE
peer state = 8  -STANDBY HOT
Mode = Duplex
Unit = Secondary
Unit ID = 2
```

```
Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured)  = Stateful Switchover
Redundancy State               = Stateful Switchover
                               <snip>
```

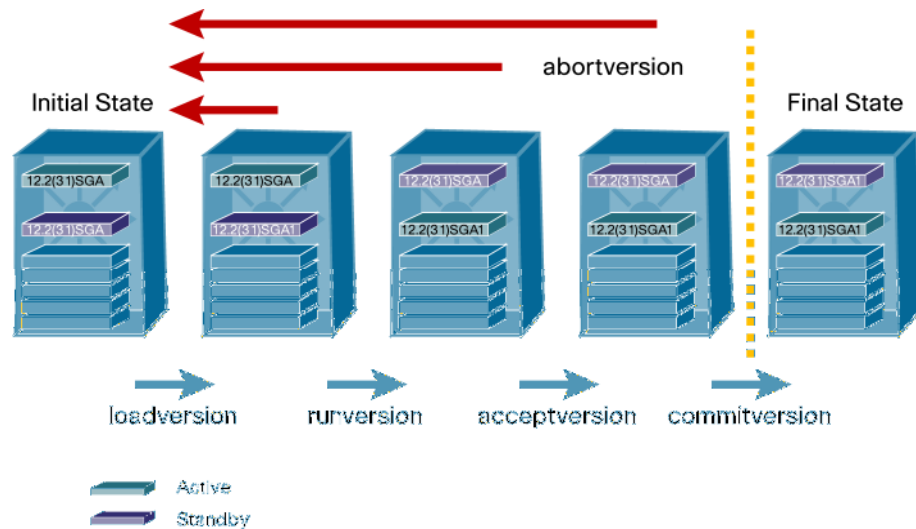
```
4507R-ISSU#sh run | b redundancy
```

```
redundancy
mode sso
```

ISSU Process

The ISSU process consists of several steps (Figure 3). Each step is explicitly initiated by the user by invoking a specific CLI command. Throughout the process, the user has the ability to make sure and verify that there is no degradation of the service. A CLI command is available to manually abort the entire process. Figure 3 summarizes these steps. Further details of each step will be explained in greater detail in the subsequent section of this document.

Figure 3. Steps to Perform an ISSU Upgrade from Cisco IOS Software Release 12.2(31)SGA to Release 12.2(31)SGA1



As a step prior to the beginning of the ISSU process, the new version of the Cisco IOS Software image needs to be loaded into both the active and standby supervisors' file systems. Both active and standby supervisor need to contain both the new and old images in the file system. In order to store both new and old images, the supervisors should be upgraded to contain sufficient amounts of flash memory prior to the ISSU process.

The new images can be downloaded into both supervisors using commands such as:

```
copy tftp: bootflash:
copy tftp: slavebootflash:
```

The example below illustrates this verification:

```
4510R-203#dir
Directory of bootflash:/

 1  -rwx 13636500 Sep 6 2006 03:18:58 -08:00 cat4500-entservices-mz.122-31.SGA
 2  -rwx 13747611 Sep 9 2006 03:19:58 -08:00 cat4500-entservices-mz.122-31.SGA1

4510R-203#dir slavebootflash:
Directory of slavebootflash:/
```

```

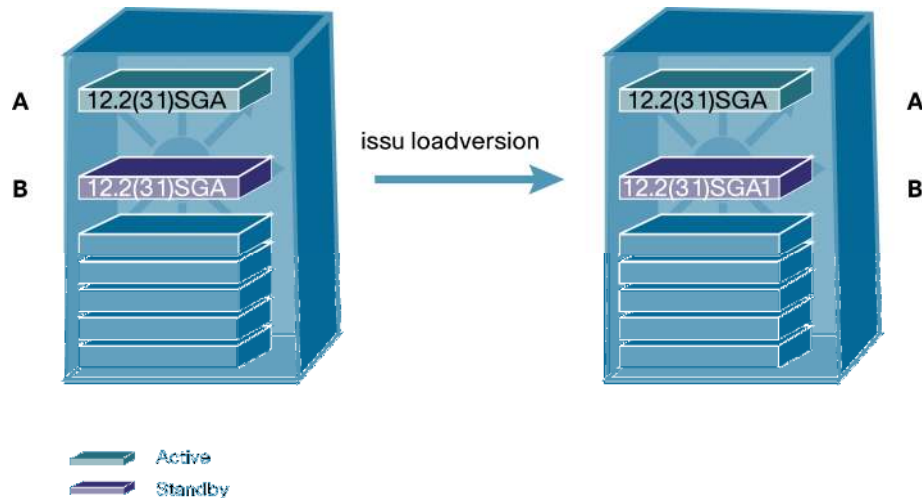
1 -rwx 13636500 Sep 6 2006 03:18:58 -08:00 cat4500-entservices-mz.122-31.SGA
2 -rwx 13747611 Sep 9 2006 03:19:58 -08:00 cat4500-entservices-mz.122-31.SGA1

```

Once this check is verified, one can now proceed with the ISSU process.

Figure 4 depicts an example of a redundant system migrating from Cisco IOS Software Release (12.2(31)\SGA to the newer Release 12.2(31)SGA1. Slot “A” represents the active supervisor, and slot “B” represents the standby.

Figure 4. Step 1: ISSU Loadversion



Note: The supervisor slots are labeled “A” and “B” throughout the illustrations in this paper. In reality, supervisor slots on a redundant Cisco Catalyst 4500 are slots 1 and 2, respectively.

The ISSU process is started by typing the “issu loadversion” command on the active supervisor. This command directs the active supervisor to begin the ISSU process. The active supervisor, through intersupervisor communications, checks that the requested image has been downloaded into both the active and standby supervisors’ file systems. If the required images are not present, the command is rejected, and an appropriate warning is generated.

If the “issu loadversion” command is successful, the switch transitions into the “Load Version” ISSU state. The standby supervisor will reset and boot with the new version of the Cisco IOS Software image loaded into the file system.

The following actions take place when the command is implemented:

1. The standby supervisor (B) is reset.
2. The standby supervisor (B) is booted with the new Cisco IOS Software image: Release 12.2(31)SGA1.
3. If both Cisco IOS Software images are declared as compatible, the standby supervisor moves into SSO mode and is fully stateful for all compatible clients and applications. Compatibility allows for in-service software upgrade or downgrade between two versions to succeed with minimal service effect.

4. If both Cisco IOS Software images are incompatible, the system moves into RPR mode, and the ISSU process is terminated with an appropriate message to the user. Images are declared incompatible when “required” clients or applications are not interoperable between two Cisco IOS Software releases.
5. Standby “B” reaches the standby HOT state.
6. The user has an option to abort the ISSU process by issuing the “issu abortversion” command.
7. The “issu loadversion” command also supports a “forced” option that allows the operator to force the system into entering RPR mode when incompatibility is detected.

Note: When performing an ISSU, disable manual switchovers. Performing manual switchovers during the issu process is strongly discouraged. The current implementation does not prevent it, but it does display a warning to the user.

An example of the CLI for implementing the issu loadversion command is displayed below.

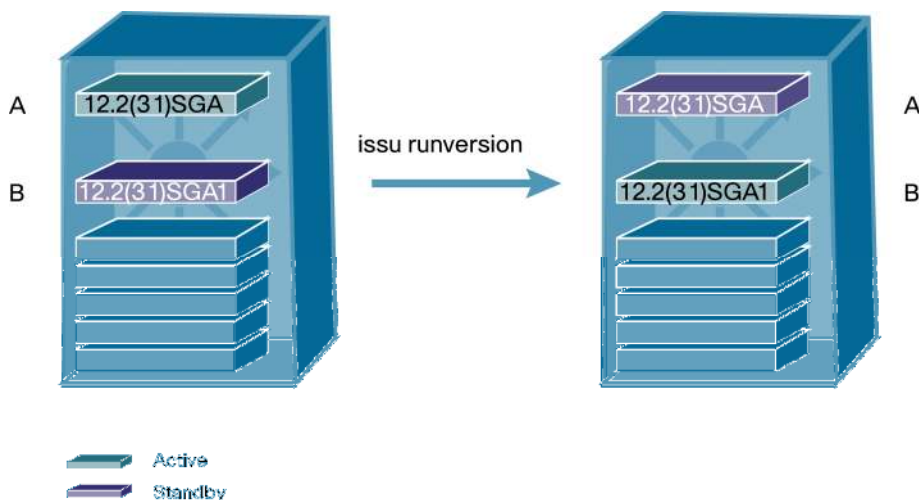
On the active supervisor, one would issue the following command:

```
4510R-203#issu loadversion 1 bootflash:cat4500-entservices-mz.122-31.SGA1 2 slavebootflash: cat4500-entservices-mz.122-31.SGA1
```

Syntax - **issu loadversion** active-slot active-image-new standby-slot standby-image-new

Figure 5 depicts an example of a stateful switchover via the issu runversion command. Slot B is now the active supervisor running the new version of Cisco IOS, while Slot A is the standby supervisor running the old version of Cisco IOS.

Figure 5. Step 2: ISSU runversion



The second step of the ISSU process is to perform the issu runversion CLI.

The user can issue the “issu runversion” command when:

1. The ISSU state is “Load Version”; this can be verified with the “show issu state detail” CLI.
2. The standby supervisor is running the new version of the software.
3. The standby supervisor has moved into the “Standby Hot ” state.

The following actions take place when the “issu runversion” command is executed:

1. A switchover occurs; that is, the standby (B) becomes the new active, and the old active (A) is rebooted and comes up as a standby.
2. A timer called “Rollback Timer” is started with a previously configured value.
3. Move both supervisors to “Run Version” state.
4. If the command “issu acceptversion” is not issued before the “Rollback timer” fires, then the entire ISSU process is aborted via the automatic rollback.
5. If the active supervisor console connectivity is established and the “issu acceptversion” command is issued, then the rollback timer is stopped.
6. The user has an option to abort the ISSU process by issuing the “issu abortversion” command.

An example of the CLI for implementing the issu runversion command is displayed below:

On the active supervisor, one would issue the following command:

```
4510R-203#issu runversion 2 slavebootflash:cat4500-entservices-mz.122-31.SGA1
Syntax - issu runversion standby-slot [standby-image-new]
```

ISSU Process: Rollback-timer

Cisco IOS Software maintains an ISSU rollback timer. The automatic rollback timer provides a safeguard against an upgrade that may leave the new active supervisor engine in a state in which communication with the standby supervisor engine is severed. For example, if you were in the middle of performing an ISSU and lost network connectivity, the rollback timer would allow for the system to revert back to its original running version.

As soon as the “issu runversion” command is issued and both supervisors initiate the transition into the “Run Version” state, a timer called the “Rollback Timer” is started. The purpose of this timer is to provide the user with a window to help ensure that the newly active supervisor is reachable both from console as well as the network. Additionally, the user can use this window of time to make sure that certain critical features and functionality are working. If the user feels that a longer window of time is needed to help ensure such functionality, the value of the timer can be extended. Once the user is satisfied that the new image is working fine, the user either issues an “issu acceptversion” command to proceed with new image or issues an “issu abortversion” command to go back to the previous version. Both commands stop the rollback timer.

The value of the rollback timer is a configurable parameter beginning from zero (meaning, disable rollback) to 2 hours; default value is 45 minutes.

```
4510R-203#sh issu rollback-timer
      Rollback Process State = In progress
      Configured Rollback Time = 45:00
      Automatic Rollback Time = 42:02

4510R-203(config)#issu set rollback-timer ?
<0-7200> Rollback timer value
```

Note: To change the default value of the rollback timer, one needs to configure the timer settings before starting the ISSU process.

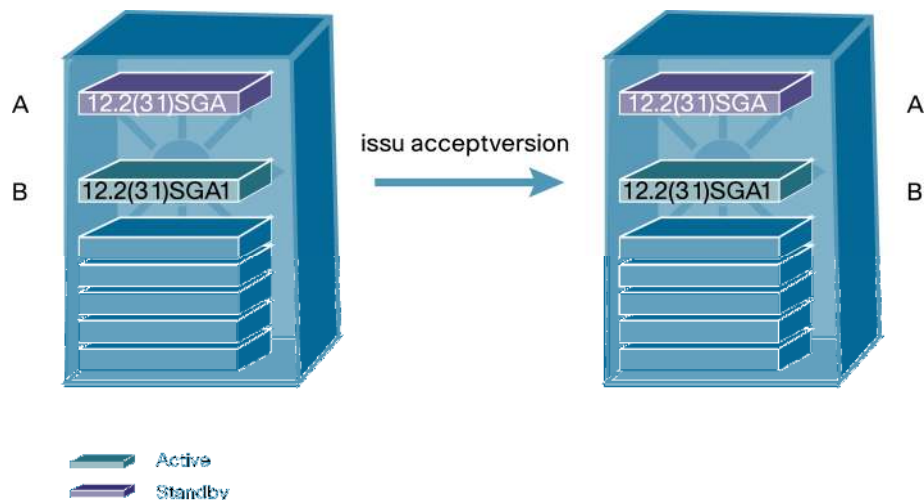
If the rollback timer expires, the ISSU process is terminated.

The following actions take place in order to prepare for and to implement rollback:

1. The ISSU state for both supervisors is set to "INIT."
2. The boot parameters of both supervisors are updated to refer to the old image version.
3. An event is logged on the active supervisor indicating that the rollback process has begun.
4. The switchover is done similar to the one mentioned in "issu runversion" with the standby supervisor becoming the active.
5. An SNMP trap is generated reporting the failure reason and previous and next states.
6. Switchovers are enabled.

Figure 6 depicts the acknowledgement of successful software activation with the `issu acceptversion` command.

Figure 6. Step 3: ISSU acceptversion



Prior to issuing the 'issu acceptversion' command the system will be counting down the rollback timer. If 'issu acceptversion' is not completed before rollback timer expires an automatic abort will occur. This command stops the "Rollback Timer." This command serves as a feedback mechanism. This is an optional command and can be skipped in the ISSU process with the "issu commitversion" CLI.

If this command is not issued within 45 minutes (default) from the time the standby supervisor moves into the "Standby Hot" state, it is assumed that the new active supervisor is not reachable and the entire ISSU process is rolled back to the previous version of the software. The acceptversion is not intended for long-term network operation. It is also important to note that none of the features available on the new version will work yet.

The following actions take place when the command is implemented:

1. The "Rollback Timer" is terminated. This means that the rollback timer is not looked at anymore. Therefore, the system can run in this state for an extended period.

-
2. The user has an option to abort the ISSU process by issuing the command "issu abortversion."

Aborting the ISSU process now causes the newly active supervisor (B) to fail over to the standby supervisor (A) running the old image and will also cause the rebooting supervisor (B) to load the original image. The `issu acceptversion` halts the rollback timer and helps ensure the ISSU process is not automatically aborted during the process.

An example of the CLI for implementing the `issu acceptversion` command is displayed below:

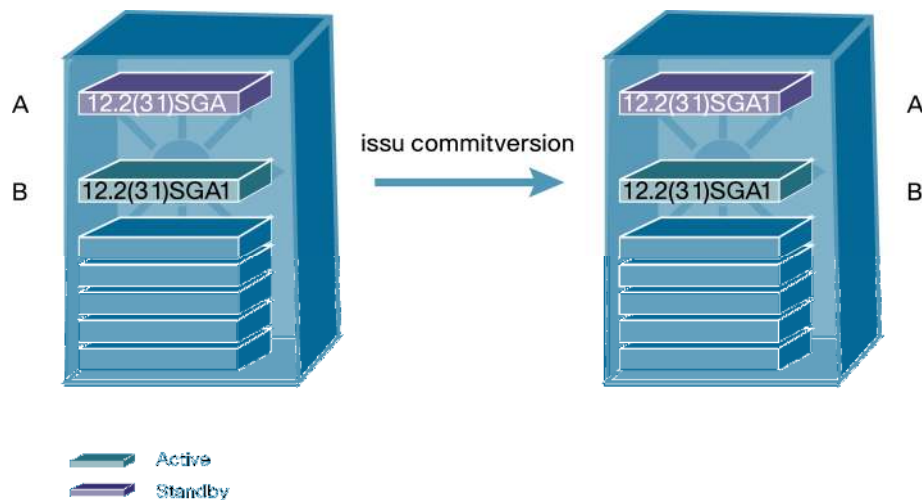
On the “New” active supervisor, one would issue the following command:

```
4510R-203#issu acceptversion 2
% Rollback timer stopped. Please issue the commitversion command.
```

```
Syntax - issu acceptversion active-slot-number
```

Figure 7 shows when committing the new Cisco IOS image on the active supervisor, the standby supervisor resets and returns with the new image.

Figure 7. Step 4: ISSU commitversion



This is the last stage of the ISSU procedure. Once the user is satisfied with the new version of software, this must be committed by issuing the “`issu commitversion`” command. This command resets the standby supervisor and boots it with a new version of the software (same as the active supervisor). This concludes the ISSU process, and the new version of software is permanently committed on both supervisors. Since this is the conclusion of the ISSU process, the system can not be reverted back to the previous version of the software from this point onward as a part of this upgrade cycle. However, if for any reason users wish to go back to the previous version of the software, they can do so by starting a new upgrade/downgrade process.

The following actions take place if the command is implemented:

1. The standby supervisor (A) is reset and booted with the new version of Cisco IOS Software image.
2. The standby supervisor (A) moves into the “Standby Hot” state in SSO mode and is fully stateful for all clients/applications that are compatible.
3. Both supervisors are moved into “Final State,” which is the same as “Initial State.”
4. Users can initiate switchovers from this point onward.

An example of the CLI for implementing the `issu commitversion` command is displayed below:

```
4510R-203#issu commitversion 1
Syntax - issu commitversion standby-slot-number
```

ISSU Process: `issu abortversion`

One can abort the ISSU process at any stage manually (prior to issuing the `issu commitversion` command) by issuing the exec-level `issu abortversion` command. The ISSU process also aborts on its own if the software detects a failure.

If a user aborts the process after issuing the `issu loadversion` command, then the standby supervisor engine is reset and reloaded with the original software.

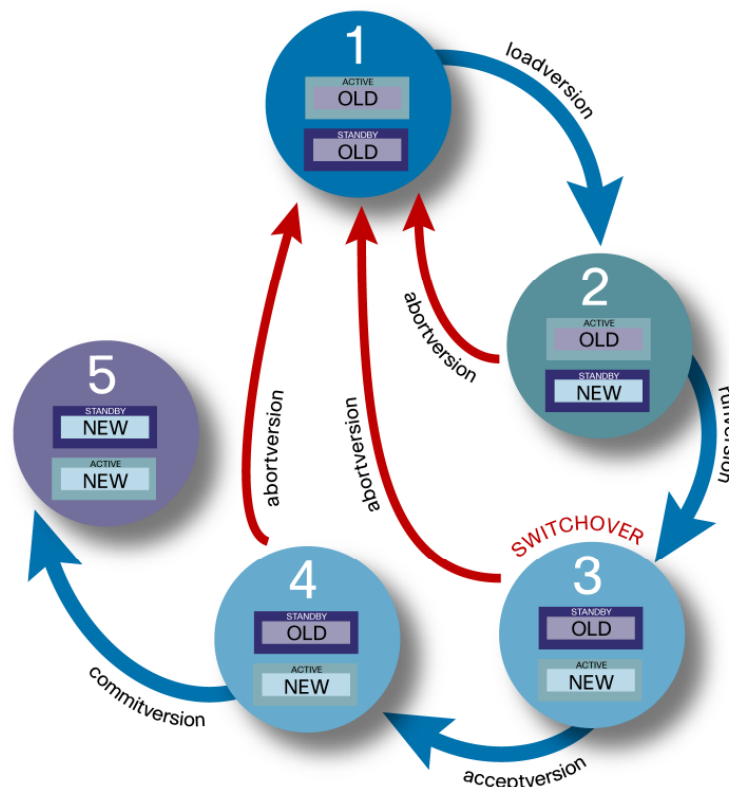
If the process is aborted after a user enters either the `issu runversion` or `issu acceptversion` command, then a second switchover is performed to the new standby supervisor engine that is still running the original software version.

The supervisor engine that had been running the new software is reset and reloaded with the original software version. The command is accepted only in “Load Version” or “Run Version” states. In “Load Version” state, the active supervisor is running an old image and the standby supervisor is running new image.

```
Syntax - issu abortversion active-slot [active-image-new]
```

Figure 8 shows how the `issu` process can be aborted at any stage prior to the `commitversion`.

Figure 8. ISSU abortversion



ISSU Requirements

- Before one performs an ISSU, make sure the system is configured for redundancy mode SSO and the file system for both the active and the standby supervisor engines contains the new ISSU-compatible image. The current version running in the system must also support ISSU. You can issue various commands on the Cisco Catalyst 4500 Series switch to determine supervisor engine versioning and compatibility.
- Do not make any hardware changes while performing an ISSU process.
- ISSU is available starting with Cisco IOS Software Release 12.2(31)SGA for the Cisco Catalyst 4500 Series.
- The new features should not be enabled (if they require change of configuration) during the ISSU process.
- In a downgrade scenario, if any feature is not available in the downgrade revision of Cisco IOS Software image, that feature should be disabled prior to initiating the ISSU process.

ISSU Guidelines

The following guidelines are recommended and should be followed in order to make the ISSU process successful.

- **No switchovers:** When the ISSU process is initiated, no manual switchovers should be attempted. The user must not force any switchovers other than switchovers initiated as a part of this process.
- **ISSU works only in SSO mode:** ISSU is a process to upgrade/downgrade an SSO compliant image from one version to another. Both versions of images must be at least base-level compatible. Additionally, both supervisors must be configured in SSO mode. If either of these conditions is not met, the ISSU process is rejected; that is, the command “issu loadversion” is rejected.
- **ISSU requires two supervisors:** ISSU requires that there exist two supervisors in a chassis that is being upgraded or downgraded.
- **Predownload of images required:** Prior to initiating the ISSU process, the new image must be downloaded in the local file system of both supervisors. If not done so, the “issu loadversion” command is rejected, and hence the ISSU process cannot be started.

Cisco Catalyst 4500 ISSU Summary

ISSU allows customers to perform full-feature Cisco IOS Software upgrades with minimal to no effect on a redundant Cisco Catalyst 4500 system. It enables rapid, nondisruptive software upgrade for new line cards, new power supplies, new features, or bug fixes. ISSU offers continuous packet forwarding during the supervisor engine switchover running different Cisco IOS Software versions. Together with NSF/SSO, ISSU allows forwarding of data packets along known routes and avoids unnecessary route flaps and network instabilities during software upgrade. As a result IP phone calls do not drop even when the Cisco IOS Software images are upgraded or downgraded. ISSU is typically deployed in the enterprise wiring closet or service provider metro Ethernet aggregation. ISSU virtually eliminates planned downtime to maximize system and network availability.

