High-Availability Solution Overview

Cisco IOS Software: Guide to Performing In-Service Software Upgrades

In most networks, a significant cause of downtime is planned maintenance and software upgrades. Recent enhancements now allow modification of Cisco IOS® Software while packet forwarding continues, reducing downtime due to planned software upgrades and increasing availability.

Enterprises, service providers, in fact nearly all network operators find that a significant percentage of downtime results from planned network maintenance. Software upgrades to implement new features or capabilities, or to apply maintenance are primary causes for system inaccessibility. Ensuring a highly available network means eliminating all possible causes of downtime, and availability has never been more important than today. The criticality of network services and necessity of access to core business applications means maintenance windows have been curtailed or even eliminated altogether. Networks now need to function 24 hours a day. But enhancements and upgrades yielding new capabilities and services, changes, and maintenance must go on.

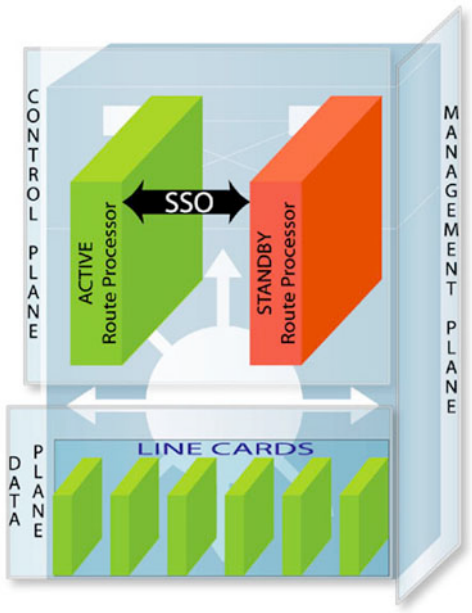
Path redundancy often is used to facilitate access during periods of maintenance activity, but single points of failure sometimes exist or the network design simply does not allow for access if a node is taken out of service. What more can be done to limit the impact of planned upgrades?

Fortunately, *recent enhancements allow modification of Cisco IOS Software while packet forwarding continues*. Cisco IOS Software Release 12.2S (first available for the Cisco® 10000 in 12.2SB, a 12.2S derivative) now supports *Cisco In-Service Software Upgrade (ISSU)*. Cisco IOS ISSU takes advantage of Cisco Nonstop Forwarding with Stateful Switchover (NSF/SSO) and hardware redundancy to permit true in-service software upgrades or version changes while continuously forwarding user traffic. Cisco IOS Software high-availability features combine to lower the impact planned maintenance activities have on network service availability. The result is less downtime and better access to critical systems – anytime, anywhere (Figure 1).

This document describes how to take advantage of this evolutionary addition to Cisco IOS Software. It explains what Cisco IOS ISSU is and how it works. Included are step-by-step instructions for performing in-service software version changes or maintenance on the Cisco 10000 Series routers. With this knowledge, you will understand how to modify your operations procedures to take full advantage of Cisco IOS ISSU and improve network availability.

Figure 1

Cisco IOS ISSU takes advantage of the Cisco IOS Software high-availability architecture, the separation of data and control planes, and Stateful Switchover.



What Is Cisco IOS ISSU?

Consider Cisco IOS ISSU as a procedure. In order to perform an upgrade while the router concurrently forwards packets, you must first have:

- A router with redundant control plane hardware (that is, redundant route processors)
- A separate data or forwarding plane
- Previous implementation of Cisco NSF/SSO

Unlike SSO, which is a mode of operation for the device and a prerequisite to Cisco IOS ISSU, the upgrade process is a series of steps that result in the implementation of new Cisco IOS Software – while the router or switch is in operation and with minimal impact on traffic.

In essence, the procedure can be simply described as follows:

1. Load new software on the standby route processor.

2. Switch over to the new software.

3. Reset the new standby route processor (the original active route processor) with the new software.

Although the process is easy from an operational perspective, it is not that simple from an engineering perspective. This upgrade is a true in-service upgrade with support for version upgrades that include new features and functions, not just simple code patches. Quite a bit of engineering, technology, infrastructure software, and an underlying architecture for high availability make it all possible. These mechanisms come together to give the network operator a simple interface to a complex process.

Note: If you are not already familiar with Cisco NSF/SSO, it might help to first study the Cisco IOS Software documentation for Stateful Switchover (SSO) and Nonstop Forwarding (NSF) and the Cisco Nonstop Forwarding with Stateful Switchover Deployment Guide.

As mentioned earlier, Cisco IOS ISSU depends on Cisco NSF/SSO, which depends on fully redundant control plane elements or route processors and a continually operating data plane. Different Cisco products use different types of hardware, and the terminology reflects this fact. For example, the route processors on the Cisco 10000 Series routers are known as performance routing engines (PREs). This document refers to route processors generically to indicate the redundant control plane hardware elements.

New Versioning Capability

Prior to the introduction of the Cisco IOS ISSU capability, the SSO mode of operation required each route processor to be running like versions of Cisco IOS Software. The operating mode of the system in a redundant high-availability configuration is determined by exchanging version strings when the standby route processor registers with the active route processor. In the past, the system entered SSO mode only if the versions running on both route processors were the same. If not, the redundancy mode was reduced to ensure compatibility. Now, with the introduction of Cisco IOS ISSU capability, the implementation allows two different but compatible Cisco IOS Software release levels to interoperate in SSO mode and allows software upgrades while packet forwarding continues.

The previous version checks are no longer sufficient to allow the system to properly determine the operating mode. Cisco IOS ISSU requires additional information to adequately determine compatibility between software versions and the included component subsystems. Therefore, a compatibility matrix is defined that contains information about other images with respect to the one in question. This matrix is necessary to correctly represent the compatibility of two software versions, one running on the active and the other on the standby route processor, and furthermore, to allow the system to determine the highest operating mode it can achieve. Incompatible versions cannot progress to SSO operating mode.

The Cisco IOS Software infrastructure has been internally modified and redesigned to accommodate client feature or subsystem message versioning. Cisco IOS Software subsystems correspond to feature sets and software component groupings. Features or subsystems that maintain state information across route processors are said to be *high-availability-aware* or *SSO clients*. A mechanism allows subsystems within Cisco IOS Software to communicate – route processor to route processor – and to negotiate the message version for communication between route processors. This mechanism is referred to as Cisco IOS ISSU Framework (or Cisco IOS ISSU protocol). Internally, all SSO-compliant applications or subsystems (those that are said to be high-availability-aware and keep state information synchronized using the checkpoint facility) must follow this protocol to establish communication with their peer across different versions of software.

Cisco IOS ISSU is designed to accommodate changes within the same Cisco IOS Software release train. Cisco IOS ISSU is available now for Release 12.2S, with initial support beginning with Cisco 10000 Series routers. In-service software upgrades between minor releases, including new technology within 12.2S is allowed (that is, 12.2(31)S to 12.2(32)S).

A brief description of Cisco IOS Software Release Trains is included in Figure 2.

Figure 2
Cisco IOS Software Release Trains

Maintenance releases consolidate the new technology introduction releases from the previous release family. They are widely deployable with broad hardware adoption and extensive application support. Maintenance releases receive software fixes regularly, but no new features or hardware support. Examples include Cisco IOS Software Major Release 12.2 and Major Release 12.3.

New Technology Introduction releases are derived from the major release that shares the same number. For example, Release 12.3T is derived from Major Release 12.3. Like maintenance releases, New Technology Introduction releases are widely deployable with broad hardware adoption and extensive application support. In addition to regular software fixes, New Technology Introduction releases provide new features and hardware support. Examples include Release 12.2S and Release 12.3T.

Application-Specific releases focus on a single technology or customer. They offer narrow hardware adoption and are intended to have a very limited life. The functions introduced in an Application-Specific release are consolidated into one of the main New Technology Introduction releases at the earliest opportunity.

For more information about Cisco IOS Software releases, refer to ABCs of Cisco IOS Software Release.

Between each minor release, Cisco testing determines the in-service upgrade and downgrade compatibility for all internal SSO-capable software functions. The testing determines Cisco IOS ISSU compatibility and designates the release in accordance with two levels of support or designates the release incapable of in-service upgrade and instead allows for a *fast software upgrade* (FSU). There exists a core set of system infrastructure software that must be able to interoperate between Cisco IOS versions in a stateful manner for SSO to function correctly. Other high-availability-aware subsystems may or may not be required from a customer viewpoint (that is, not used or configured). These can be considered *optional*.

The three Cisco IOS ISSU compatibility designations follow:

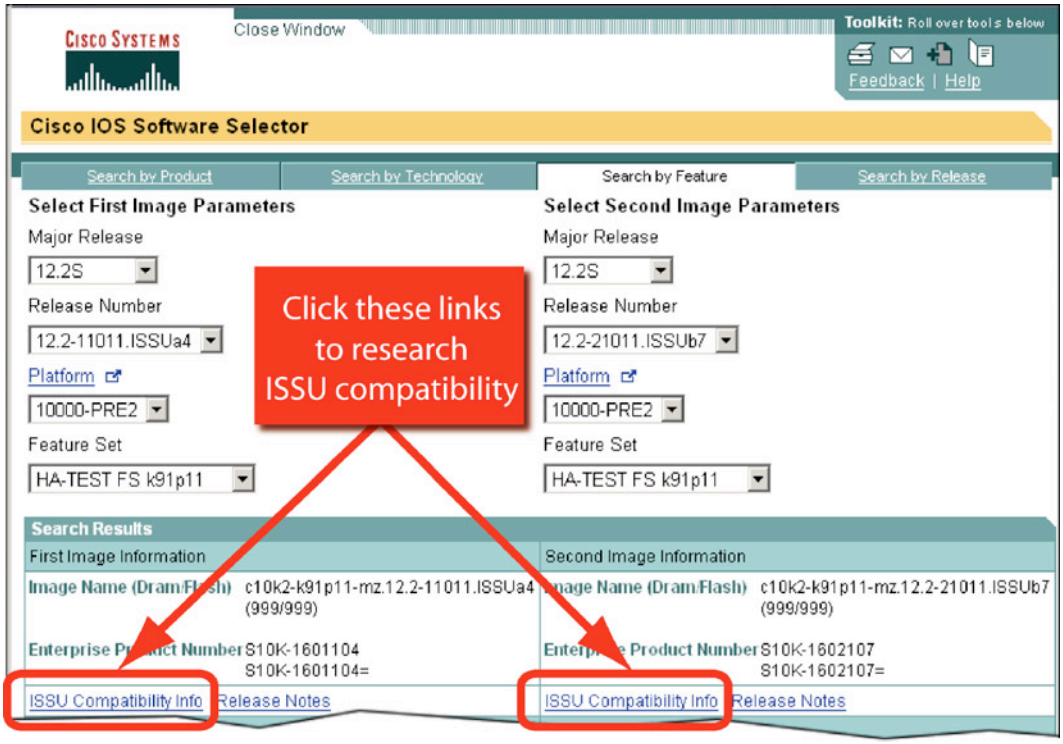
- **Compatible** – When the *base-level system infrastructure and all optional high-availability-aware features or subsystems are compatible*, the Cisco IOS Software image versions are considered "compatible." This means an in-service upgrade or downgrade between these versions will succeed with minimal service impact.
- **Base-level compatible** – If *one or more of the optional high-availability-aware subsystems are not compatible*, the Cisco IOS Software image versions are considered "base-level" compatible. This means an in-service upgrade or downgrade between these versions will succeed, but some subsystems will not be able to maintain state during the transition. Careful consideration of the impact this may have on operation and service is required before attempting an in-service upgrade.
- **Incompatible** – There exists a core set of system infrastructure software that must be able to interoperate between Cisco IOS versions in a stateful manner for SSO to function correctly. If any of these "required" features or protocols is not interoperable between two releases, then the two versions of the Cisco IOS Software images are declared "incompatible," meaning an in-service upgrade or downgrade between these versions is not possible. Alternatively, a *fast software upgrade* can be performed, but a FSU impacts service.

The compatibility matrix represents the compatibility relationship a Cisco IOS Software image has with all the other Cisco IOS Software versions that are within the designated support window (that is, all those it "knows" about) and is populated and released with every image. The matrix stores compatibility information between its own release and prior releases that exist. It is always the newest release that contains the latest information about compatibility with existing releases in the field.

Besides being contained within the Cisco IOS Software image, information from the compatibility matrix is available on Cisco.com so that customers, network designers, and planners can determine in advance whether an upgrade can be performed using the Cisco IOS ISSU process. Simply use the familiar Cisco IOS Software Feature Navigator tool and the Software Selector tool (URLs are Cisco.com/go/fn and Cisco.com/go/ciss, respectively).

From there you can navigate from the main menu to see information regarding the compatibility of each Cisco IOS Software release. For example, Figure 3 illustrates where to click to see Cisco IOS ISSU compatibility information when comparing Cisco IOS Software releases using Cisco IOS Software Selector.

Figure 3
Researching Cisco IOS ISSU Compatibility from Cisco.com



For more details about using Cisco IOS Software Feature Navigator and the Software Advisor to determine Cisco IOS ISSU compatibility, refer to the companion paper, *Researching In-Service Software Upgrade Compatibility*.

The matrix information stores the compatibility among releases based on the rules stated previously. If the images cannot interoperate, then the matrix entry designates the images incompatible (I). If the images can fully interoperate, then the matrix entry designates the images compatible (C). And finally, when all the clients required for the two images to successfully interoperate (here referred to as “base-level clients”) are compatible while one or more non-base level clients, such as Point-to-Point Protocol (PPP), for example, are incompatible, the matrix entry designates the images base-level compatible (B).

The Cisco IOS Software ISSU compatibility matrix is visible on Cisco.com and present within every Cisco IOS Software image that supports this capability.

Note: Cisco IOS ISSU is possible only when the Cisco IOS Software on both the active and standby route processors is capable of Cisco IOS ISSU. You cannot perform an in-service software upgrade from a Cisco IOS Software version that did not support this capability to a Cisco IOS Software version that does support this capability.

Cisco Systems® performs system testing to determine Cisco IOS ISSU compatibility and conformance. Cisco back-end software build, test, and Web development systems and processes have been integrated to allow automatic preparation of Cisco IOS ISSU compatibility information to ensure the accuracy of the data. Each subsequent software image is verified against previous images. The Cisco IOS ISSU compatibility negotiation data is collected and analyzed. The data posted on Cisco.com and present within the Cisco IOS Software images comes directly from these test results.

The build, test, and data preparation process is depicted in Figure 4. Refer to the diagram as the steps are described.

1. The process begins when a new Cisco IOS Software release is built.

2. Software destined for release to customers goes through a series of tests. One suite of the tests concerns high availability and Cisco IOS ISSU verification. During the Cisco IOS ISSU compatibility testing, the compatibility matrix (CM) build process is invoked.

3. Output from the **`show issu compatibility negotiated`** command is captured and sent to the Compatibility Matrix Generator tool.

4. The Compatibility Matrix tool analyzes the output and creates a file in Extensible Markup Language (XML) format.

5. The Matrix Data tool reads the compatibility matrix XML file. This tool performs two tasks: It prepares a "C" file that is sent to the Cisco IOS Software repository to be included in the actual, finished software image, and it also prepares data for inclusion on the Cisco.com Webpages that display Cisco IOS ISSU compatibility information.

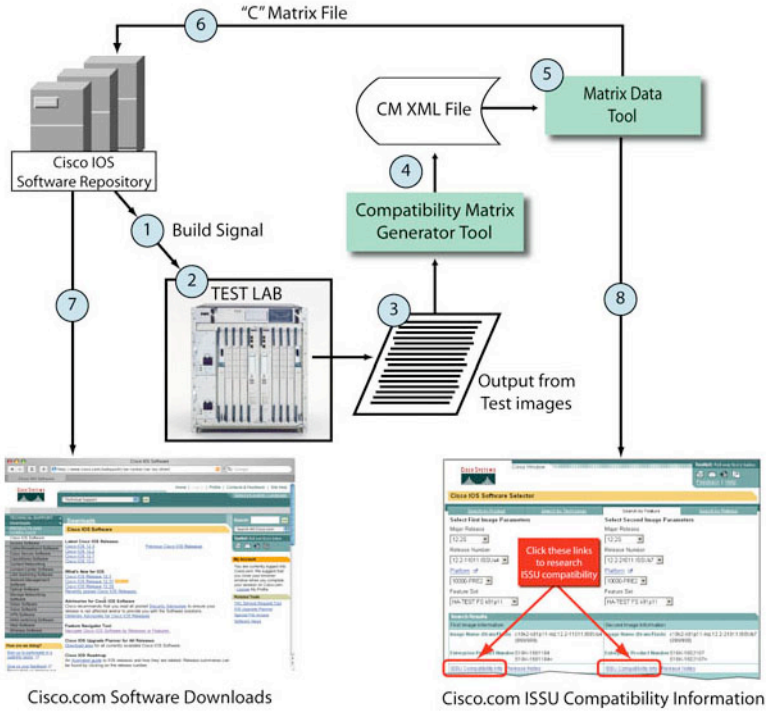
6. The "C" matrix file is stored in the Cisco IOS Software repository.

7. The final Cisco IOS Software images are posted to the Cisco.com download area.

8. The Cisco IOS ISSU compatibility information is posted in Cisco IOS Software Selector and Feature Navigator on Cisco.com.

Figure 4

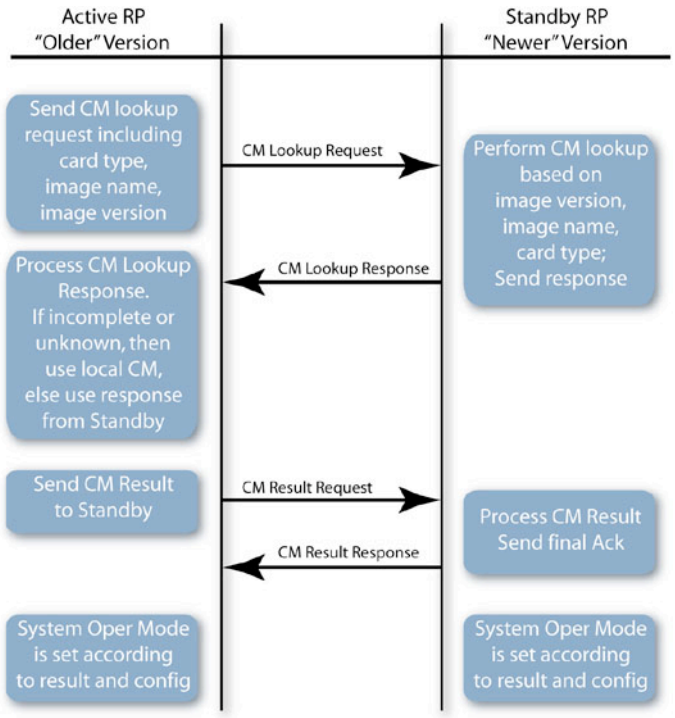
Cisco IOS ISSU Compatibility Verification and Build Process



Before attempting an in-service software upgrade, you should know the compatibility level between the two Cisco IOS Software versions in question. You can research Cisco IOS ISSU compatibility using the tools available on Cisco.com. The tools provide accurate data for planning purposes prior to actually performing an upgrade.

As mentioned previously, when redundant systems are brought up, the stored compatibility matrix information is analyzed internally to determine interoperability between the software running on the active and standby route processors. This analysis can happen only when the software on both the active and standby route processors supports Cisco IOS ISSU. It also occurs during the Cisco IOS ISSU process as the Cisco IOS ISSU commands are issued and the redundant route processors are reset and reloaded. The compatibility matrix analysis is part of the criteria used to set the operating mode of the system. The data exchange between the two redundant route processors is illustrated in Figure 5.

Figure 5
Cisco IOS Software Compatibility Negotiation



The Cisco IOS Software high-availability infrastructure and architecture provide that all stateful client software components negotiate their registered messages with their peer endpoints to determine the message exchange version to be used during state maintenance. In addition, all clients negotiate any unique capabilities with their peer endpoint on the standby route processor. When this process concludes, a final determination of the system operating mode is made.

Cisco IOS Software subsystems that are high-availability-aware and synchronize state information are uniquely identified by a global scope ID (that is, client ID) within the Cisco IOS ISSU infrastructure. The actual negotiation of the compatibility matrix data between two software versions on a given system can be displayed using the following command-line interface (CLI) commands:

```
show issu comp-matrix stored

show issu comp-matrix negotiated
```

The option **stored** is used to display the compatibility matrix information stored within the Cisco IOS Software image running local to where the command is issued. The option **negotiated** is used to display the negotiated compatibility matrix information between two different Cisco IOS Software versions running on each of the two route processors. This output is used during Cisco compatibility verification and testing to create the compatibility matrix information that is available on Cisco.com as well as compiled within the Cisco IOS Software.

Cisco IOS ISSU Version Compatibility Expectations

Although the outcome of testing and the subsequent generation of the compatibility matrix establish the actual support for in-service upgrade between specific Cisco IOS Software releases, the general expectations include the following:

1. In-service upgrade (or downgrade) from one minor release to another is allowed; for example, from 12.2(28)S to 12.2(29)S.

2. Upgrade (or downgrade) between minor releases not in sequence is also allowed. In other words, you can skip minor maintenance releases; for example, 12.2(28)S to 12.2(31)S. The span of releases having Cisco IOS ISSU support is expected to cover a rolling time window of approximately 18 months.

3. In-service upgrade (or downgrade) from one minor release rebuild to another is allowed; for example, from 12.2(29)S1 to 12.2(29)S3.

4. Upgrade or downgrade across major releases may not be supported. Major release changes might be when base-level client (clients required for the two systems to interoperate correctly) changes are implemented such that in-service upgrades would not be allowed. That said, Cisco IOS Software Release 12.2S is expected to continue along with regular minor releases for at least the next few years, so you can expect to gain significant benefits from Cisco IOS ISSU.

5. Upgrades and downgrades are possible only within the given major release train. This means crossing between S and T or Mainline release is not typically possible even if each release train has Cisco IOS ISSU capability.

Note: Here, the term “minor” release is synonymous with maintenance release but used in the context of a new technology release. An overview of the Cisco IOS Software release process and the definition of release terminology is available at: http://www.cisco.com/en/US/partner/products/sw/iosswrel/ios_abcs_ios_networking_the_enterprise_listing.html.

As mentioned previously, the compatibility matrix data that results from Cisco testing is stored and shipped with each Cisco IOS Software image that supports the Cisco IOS ISSU capability.

Minimum Disruption Restart for Line Cards

Distributed routing and switching systems scale in performance and capacity through the use of special hardware, processors, and software present on line cards or modules. Even centralized switching products can accommodate intelligent line cards with their own operating systems. Products such as the Cisco 10000 that have intelligent line cards support Minimum Disruption Restart (MDR) to facilitate Cisco IOS ISSU with minimal impact to data forwarding. MDR is used to load new line-card images, when necessary, with only a brief data plane outage, minimizing the impact to traffic as much as possible.

The level of support for MDR varies, depending on the product, the architecture, and the line cards themselves. Because Cisco IOS ISSU is first available on the Cisco 10000, this document examines that product more closely.

Note: MDR is invoked automatically as part of the Cisco IOS ISSU process. It does not need to be enabled or configured.

Cisco 10000 MDR

The line cards on the Cisco 10000 can represent single points of failure. Even when route processor (dual PRE-2) redundancy exists and a Cisco IOS ISSU is being performed, the line-card software upgrade can impact network traffic flow. Line-card upgrades using MDR minimize the impact on traffic flow such that service remains unaffected.

Generically, the steps involved in the in-service upgrade of a nonredundant component such as a line card follow:

1. Determine if the component software needs to be changed.

2. Initiate the MDR reload of the component, which loads the new up-level or down-level image into memory while leaving the card-level control plane active.

3. Allow the new image to initialize, while continuing to leave the control plane active.

4. Wait for the new image to synchronize with the stored state information from the old image.

5. If successful, commit the new image. Otherwise, roll back to the original image and abort.

The Cisco 10000 uses a centralized forwarding architecture. However, the line cards run an embedded operating system known as Cisco Line Card Operating System (LCDOS). In order to perform a Cisco IOS ISSU on a nonredundant line card with minimal session or packet loss, the new LCDOS image must be loaded into the line-card memory, and then the line-card control processor must initialize and run the new image in as short a time as possible – all while maintaining the integrity of the data plane. Actual switching of traffic halts briefly and then resumes following reconnect with the forwarding processor on the PRE. To maintain the data plane while reloading the card-level control plane, all hardware in the data plane must remain intact and cannot be reset or modified in any way that stops forwarding data for a significant period of time.

The MDR implementation on the Cisco 10000 achieves this scenario by loading and restarting the line-card software with only minor interruption to the flow of traffic on the data plane. Again, this occurs automatically as part of the Cisco IOS ISSU process.

Configuration Synchronization

As previously mentioned, SSO is a prerequisite for Cisco IOS ISSU. Two important benefits are gained by a system operating in SSO mode:

- Network applications (Cisco IOS Software features or subsystems that are high-availability-aware clients) synchronize their dynamic state information from the active route processor to the standby route processor.
- The persistent data of the network application (that is, its configuration information) is kept consistent between the active route processor and the standby route processor.

Cisco IOS ISSU Configuration Synchronization (Config Sync) is a method and service within Cisco IOS Software that synchronizes CLI configuration commands between the active and standby route processors. Prior to Cisco IOS ISSU, the software versions running on the active and standby route processors were always the same, so the same configuration commands were always supported on both the active and standby route processors.

To support Cisco IOS ISSU, Config Sync must now be able to synchronize configuration information even when one route processor is running one version of Cisco IOS Software and the redundant route processor is running another version of Cisco IOS Software. This will be the case while a Cisco IOS ISSU is in progress (after the issuance of the **`issu loadversion`** command and prior to the issuance of the **`issu commitversion`** command, as shown in the section, “Cisco IOS ISSU Procedure”).

This condition, where the active route processor is running one version and the standby route processor is running another, is allowed to persist for some time. The duration may be long or short, depending on the operational policies you put into practice for your organization.

Note: Refer to the "Best Practices for Cisco IOS Software ISSU" section for more details regarding operational policies and suggestions.

While in SSO mode with different Cisco IOS Software versions present, any configuration changes must be synchronized so the standby route processor can take over if the active route processor fails. If a particular configuration command is different between Cisco IOS Software versions, or not present at all in one of them, a dilemma exists. Obviously, a configuration that enables a new function in a newer version of software will not be supported by the older version of software, so the command cannot be synchronized to the standby route processor in this case. Any function enabled by the new configuration command would, therefore, be lost if a failure to the active route processor occurred in this situation.

Today, Cisco IOS ISSU Config Sync restricts commands that are different in Cisco IOS Software versions running on the active and standby route processors. Any attempt to enter a CLI configuration command that is determined to fail the syntax check performed on the standby route processor, results in an error message and is not allowed.

Cisco IOS ISSU Procedure

An in-service Cisco IOS Software upgrade or downgrade is accomplished by entering a series of four commands.

The command sequence is as follows:

1. `issu loadversion`
2. `issu runversion`
3. `issu acceptversion`
4. `issu commitversion`

The globes in Figure 6 represent the different states of the route processors (active or standby) and the Cisco IOS Software versions (new or old) as each of these commands is issued during the Cisco IOS ISSU process. The section "Detailed Step-by-Step Procedure" walks through each of the states as the commands are issued.

The process can be aborted at any stage. The process also aborts on its own if the software detects a failure. To manually abort the process, enter:

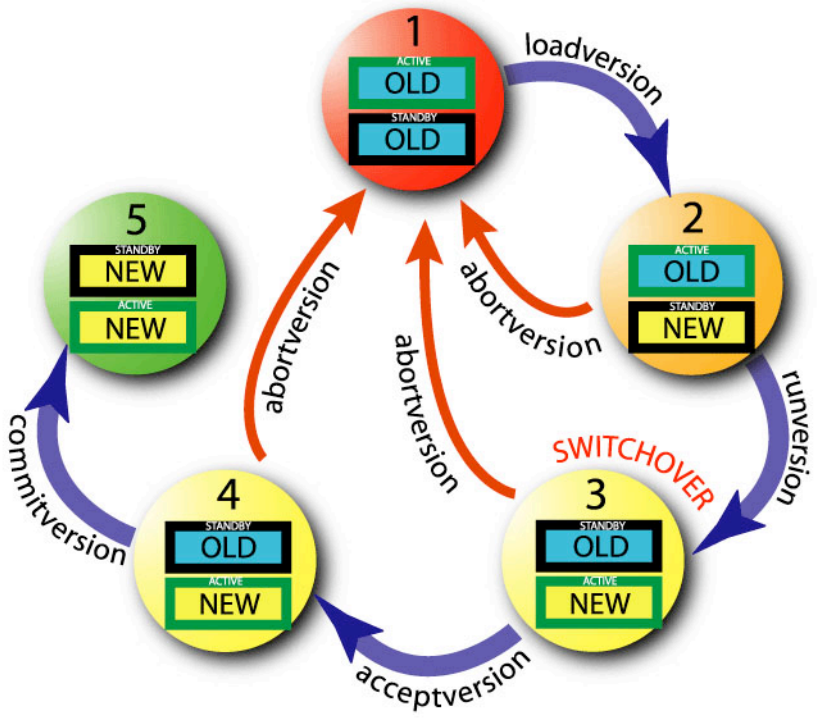
```
issu abortversion
```

If the process is aborted after the `issu loadversion` command has been issued, then the standby route processor is reset and is reloaded with the original software version. This effectively brings the state back to that in state 1 in Figure 6.

If the process is aborted after the `issu runversion` command is issued, then a second switchover is performed to the new standby that is still running the original software version. Subsequently, the route processor that had been running the new software is reset and reloaded with the original software version. Again, the state reverts to that depicted in state 1 in Figure 6.

If the process is aborted after the `issu acceptversion` command is issued, the same thing takes place as an abort after `issu runversion` – a second switchover is performed to the new standby route processor that is still running the original software version; the route processor that had been running the new software is reset and reloaded with the original software version. Again, the state reverts to that depicted in state 1 in Figure 6.

Figure 6
Cisco IOS ISSU Commands and Process



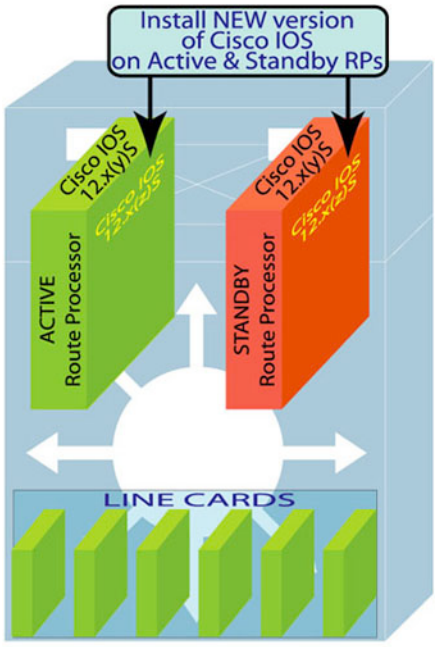
Detailed Step-by-Step Procedure

Each step of the Cisco IOS ISSU process is described in detail as follows (Figure 7):

Step 1. Copy the new version onto the active and standby route processors.

Download the new Cisco IOS Software image into both active and standby route processor file systems. This step is a prerequisite to actually initiating the Cisco IOS ISSU process.

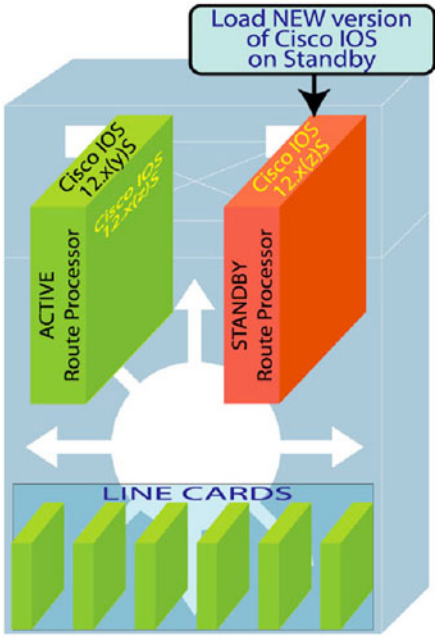
Figure 7
Copy New Version to Route Processors



Step 2. Begin the process by loading the standby route processor with the new Cisco IOS Software version (Figure 8).

Issue the `issu loadversion` command at the CLI. This command directs the active route processor to begin the Cisco IOS ISSU process. The active route processor, through interroute processor communications, helps ensure that the requested image has been downloaded into both active and standby route processor file systems and that both route processors are configured in SSO mode. The system may also perform any additional product-specific, preprocess checks at this time. If any conditions are found that would preclude continuation of the process, the command is rejected and an appropriate warning may be generated. If the internal checks pass, the standby route processor resets and boots using the new Cisco IOS Software version.

Figure 8
Load New Software on Standby



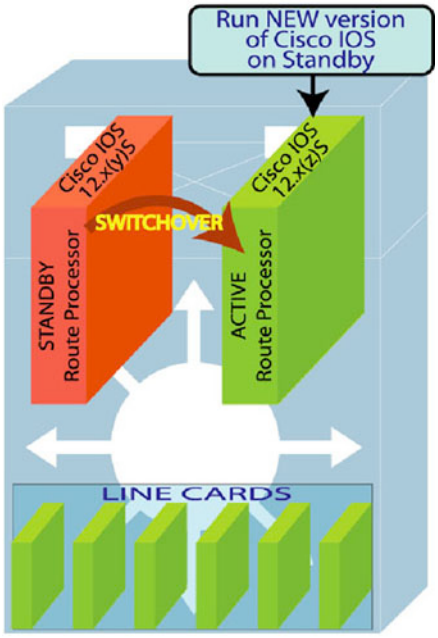
Step 3. Verify SSO status.

Ensure the standby route processor has successfully moved into the hot standby state and is running the new version of the software by issuing the `show redundancy status` command on the active route processor.

Step 4. Switch to the new version (Figure 9).

When you are satisfied that the standby route processor is "hot" and ready to take over, issue the `issu runversion` command. This command forces a route processor switchover to the standby route processor, which is now running the new Cisco IOS Software version. The standby route processor then becomes the new active route processor and NSF procedures, if configured and enabled as recommended, are performed. Packet forwarding is still handled by the data plane. The previously active route processor is reset and becomes the standby.

Figure 9
Switchover to New Software



Step 5. Verify SSO status and operational condition.

The new active route processor is running the new version of the software. At this point ensure that:

- Active route processor has console connectivity.
- Active route processor has network connectivity.
- The network has reestablished communications with routing peers and completed NSF procedures.
- The standby route processor has moved into the hot standby state and is still running the older version of the software.

Step 6. Acknowledge successful software activation.

The software maintains a timer called the Cisco IOS ISSU rollback timer. When you are satisfied that the process has been successful and wish to remain in the current state, you must indicate acceptance by issuing the `issu acceptversion` command. This command is a way for the user to give feedback to the Cisco IOS ISSU process and acknowledge successful software activation. Issuing the `issu acceptversion` command stops the rollback timer, providing a safeguard against an upgrade that may leave the new active route processor hung in a state where communication with the route processor is severed. If this command is not issued within 45 minutes (default) from the time the system has resumed SSO mode and "standby hot," it is assumed that the new active route processor is not reachable and the entire Cisco IOS ISSU process is automatically rolled back to the previous version of the software. Therefore, this command is extremely important to move the Cisco IOS ISSU process forward. Issuing the command `issu commitversion` at this stage is functionally equivalent to entering both the `issu acceptversion` *and* the `issu commitversion` commands. Use `issu commitversion` if you do not intend to run in the current state for a period of time and are satisfied with the new software version.

Note: The amount of time to wait before rolling back to the previous version is configurable using the `issu set rollback-timer <timeout value in seconds>` configuration command. The time it takes for the standby route processor to become "hot" and ready may differ, depending on product and configuration. A good tuning strategy is to set the rollback timer to several minutes greater than the amount of

time it takes for the standby to become hot and routing protocol NSF procedures and convergence to complete in your specific environment. Leave enough time to enter the `issu acceptversion` command.

Step 7. Perform full system verification.

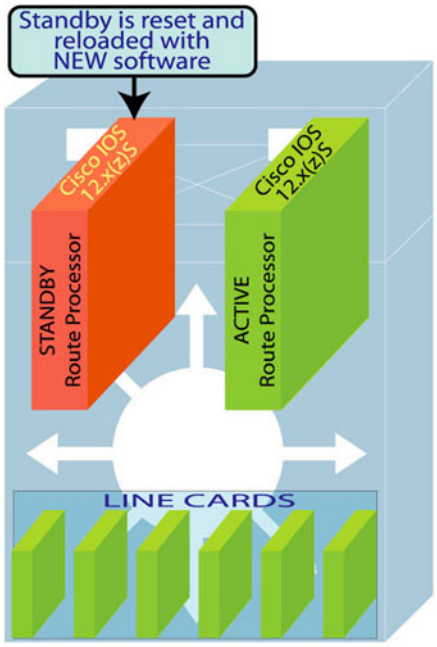
At this point, the active route processor is running the new version of Cisco IOS Software and the standby is running the previous version (assuming you have issued only the **issu acceptversion** command and not the **issu commitversion** command in step 6). This condition provides a safeguard and automatic rollback capability in case the new software version encounters some unforeseen problem. A problem in the new software that causes a route processor failure will result in an immediate switchover and rollback to the previous Cisco IOS Software version. Verification of functions while in this state is strongly recommended. Although the system can run in this state indefinitely, verification of the new image should be done as soon as possible and the system moved to the final stage of the Cisco IOS ISSU procedure. Up to this point, if for any reason you want to revert back to the older version of the software, you can issue the abort command `issu abortversion`.

Step 8. Commit the new Cisco IOS Software version (Figure 10).

When you are satisfied with the new version of software, commit the code version by issuing the `issu commitversion` command. This command resets the standby route processor and boots it with the new (currently active) version of the software. This concludes the Cisco IOS ISSU procedure and the new version of software is permanently committed on both route processors. Because this is the conclusion of the Cisco IOS ISSU process, the system cannot now revert back to the older version of the software. However, if, for any reason, you want to go back to the older version of the software, you can begin a new Cisco IOS ISSU procedure to downgrade the software.

Note: During the Cisco IOS ISSU process, you can enter the `show issu` command at any time to see the status of the process.

Figure 10
Standby is Reset and Reloaded with New Software



Automatic Abort for Cisco IOS ISSU Process

If the system determines for itself that the availability will degrade below the intended SSO mode during the Cisco IOS ISSU process, the process is automatically aborted. This helps ensure that the system is kept at the highest availability level and is prepared if failure occurs. An automatic abort returns the system to the state prior to initiation of the Cisco IOS ISSU process. An automatic abort occurs in the following cases:

- An upgrade is attempted to an incompatible Cisco IOS Software release that causes the system to become nonstateful (unable to achieve SSO mode)
- An upgrade to a compatible Cisco IOS Software release is attempted, but one or more configuration commands from the running configuration of the active route processor cannot be understood by (that is, fails a syntax check on) the standby route processor during the initial configuration bulk synchronization

Note: During normal operation, commands that fail the syntax check on the standby route processor are simply rejected on the active route processor and are not allowed to execute there. This does not hamper system redundancy.

The automatic abort is a safety mechanism. An automatic abort follows the `issu loadversion` command if something is awry. It is not expected to occur as long as an upgrade to a compatible image is being performed. If it does, it signals a problem that should be reported.

Network Management

Cisco IOS ISSU changes the operational procedures for performing Cisco IOS Software upgrades and minimizes the impact upgrades have on service. The changed operational procedure and Cisco IOS ISSU state progression can be monitored using a set of available management capabilities. Enhancements have been included to emit syslog messages and Simple Network Management Protocol (SNMP) traps or informs and to allow for interrogation of SNMP management information-based variables to provide status of the Cisco IOS ISSU process.

New Syslog Events

The following new syslog events occur for the Cisco IOS ISSU process:

- A new syslog event is logged upon execution of each of the following CLI commands:
  - issu loadversion
  - issu runversion
  - issu commitversion
  - issu abortversion
  - issu acceptversion
  - issu set rollback timer
- A new event is logged when the “rollback process” is initiated.

An example of the messages you see during an ISSU process is shown below.

Following “issu loadversion” command:

Aug 22 10:32:22.964: %ISSU_PROCESS-7-DEBUG: Peer state is [STANDBY HOT]; Please issue the runversion command

Following “issu runversion” command:

Aug 22 10:43:48.398: %ISSU_PROCESS-7-DEBUG: Peer state is [STANDBY HOT]; Please issue the acceptversion command

Following "issu acceptversion" command (console message):

% Rollback timer stopped. Please issue the commitversion command.

Following "issu commitversion" command:

Aug 22 10:51:05.520: %ISSU_PROCESS-7-DEBUG: Peer state is [STANDBY HOT]

Following "issu abortversion" command:

Aug 22 11:14:03.434: %ISSU_PROCESS-7-DEBUG: Peer state is [STANDBY HOT]

Following "issu set rollback-timer" command (console message):

% Rollback timer value set to [2000] seconds

Note: These examples were taken from prereleased software and may differ slightly from the actual messages you see.

SNMP MIB Objects

A trap indicating the progression throughout the Cisco IOS ISSU process is generated when the **issu** commands successfully complete. This trap reports the current (new) image and the previous (old) image versions. An example of the traps observed from a network management system follows:

```
- Warning Mon Aug 08 16:11:34  88.1.11.32            ISSU State Event:
ciscoRFMIBNotificationsPrefix.0.3 .1.3.6.1.4.1.9.9.176.2.0.3 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuStateRev1.0
(Integer): loadVersion  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuFromVersion.0
(OctetString): disk0:c10k2-p11-mz.CSCsb18906_minor  [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuToVersion.0
(OctetString): disk0:c10k2-p11-mz.CSCsb18906_base  [4]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusLastSwactReasonCod
e.0 (Integer): userInitiated
- Warning Mon Aug 08 16:13:14  88.1.11.32            RF Progression event:
ciscoRFMIBNotificationsPrefix .1.3.6.1.4.1.9.9.176.2 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitId.0
(Integer): 1  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitState.0
(Integer): active  [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitId.0
(Integer): 0  [4]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitState.0
(Integer): standbyCold
- Warning Mon Aug 08 16:13:39  88.1.11.32            RF Progression event:
ciscoRFMIBNotificationsPrefix .1.3.6.1.4.1.9.9.176.2 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitId.0
(Integer): 1  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitState.0
(Integer): active  [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitId.0
(Integer): 0  [4]
```

```
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitState.0
(Integer): standbyHot
- Warning Mon Aug 08 16:13:40  88.1.11.32            ISSU State Event:
ciscoRFMIBNotificationsPrefix.0.3 .1.3.6.1.4.1.9.9.176.2.0.3 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuStateRev1.0
(Integer): loadVersion  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuFromVersion.0
(OctetString): disk0:c10k2-p11-mz.CSCsb18906_minor  [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuToVersion.0
(OctetString): disk0:c10k2-p11-mz.CSCsb18906_base  [4]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusLastSwactReasonCod
e.0 (Integer): userInitiated
- Warning Mon Aug 08 16:18:19  88.1.11.32            ISSU State Event:
ciscoRFMIBNotificationsPrefix.0.3 .1.3.6.1.4.1.9.9.176.2.0.3 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuStateRev1.0
(Integer): runVersion  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuFromVersion.0
(OctetString): disk0:c10k2-p11-mz.CSCsb18906_minor  [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuToVersion.0
(OctetString): disk0:c10k2-p11-mz.CSCsb18906_base  [4]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusLastSwactReasonCod
e.0 (Integer): userInitiated
- Warning Mon Aug 08 16:20:04  88.1.11.32            RF Progression event:
ciscoRFMIBNotificationsPrefix .1.3.6.1.4.1.9.9.176.2 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitId.0
(Integer): 0  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitState.0
(Integer): active  [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitId.0
(Integer): 1  [4]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitState.0
(Integer): standbyCold
- Warning Mon Aug 08 16:20:29  88.1.11.32            RF Progression event:
ciscoRFMIBNotificationsPrefix .1.3.6.1.4.1.9.9.176.2 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitId.0
(Integer): 0  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitState.0
(Integer): active  [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitId.0
(Integer): 1  [4]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitState.0
(Integer): standbyHot
- Warning Mon Aug 08 16:20:30  88.1.11.32            ISSU State Event:
ciscoRFMIBNotificationsPrefix.0.3 .1.3.6.1.4.1.9.9.176.2.0.3 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuStateRev1.0
(Integer): runVersion  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuFromVersion.0
(OctetString): disk0:c10k2-p11-mz.CSCsb18906_minor  [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuToVersion.0
(OctetString): disk0:c10k2-p11-mz.CSCsb18906_base  [4]
```

```
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusLastSwactReasonCod
e.0 (Integer): userInitiated
- Warning Mon Aug 08 16:22:24  88.1.11.32            ISSU State Event:
ciscoRFMIBNotificationsPrefix.0.3 .1.3.6.1.4.1.9.9.176.2.0.3 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuStateRev1.0
(Integer): commitVersion  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuFromVersion.0
(OctetString): disk0:c10k2-p11-mz.CSCsb18906_minor  [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuToVersion.0
(OctetString): disk0:c10k2-p11-mz.CSCsb18906_base  [4]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusLastSwactReasonCod
e.0 (Integer): userInitiated
- Warning Mon Aug 08 16:24:15  88.1.11.32            RF Progression event:
ciscoRFMIBNotificationsPrefix .1.3.6.1.4.1.9.9.176.2 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitId.0
(Integer): 0  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitState.0
(Integer): active  [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitId.0
(Integer): 1  [4]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitState.0
(Integer): standbyCold
- Warning Mon Aug 08 16:24:40  88.1.11.32            RF Progression event:
ciscoRFMIBNotificationsPrefix .1.3.6.1.4.1.9.9.176.2 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitId.0
(Integer): 0  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusUnitState.0
(Integer): active  [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitId.0
(Integer): 1  [4]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusPeerUnitState.0
(Integer): standbyHot
- Warning Mon Aug 08 16:24:44  88.1.11.32            ISSU State Event:
ciscoRFMIBNotificationsPrefix.0.3 .1.3.6.1.4.1.9.9.176.2.0.3 VarBinds: [1]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuStateRev1.0
(Integer): init  [2]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuFromVersion.0
(OctetString):   [3]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusIssuToVersion.0
(OctetString):   [4]
private.enterprises.cisco.ciscoMgmt.ciscoRFMIB.ciscoRFMIBObjects.cRFStatus.cRFStatusLastSwactReasonCod
e.0 (Integer): userInitiated
```

Please refer to the SNMP MIB, `CISCO-RF-MIB.my` for more details.

Upgrading to an Incompatible Image

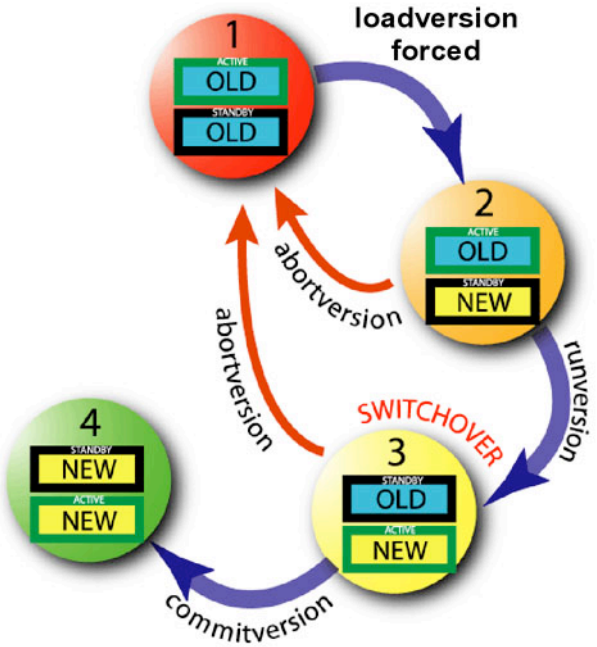
There may be times when you want to upgrade to a new Cisco IOS Software release and the new version is not Cisco IOS ISSU-compatible. Perhaps the target version has been marked "I" (incompatible), but you still want to perform an upgrade and plan for the additional service impact. An option is available within the Cisco IOS ISSU command structure to allow an upgrade without automatically aborting the process for situations when you intentionally want to drop to route processor redundancy (RPR) mode and upgrade using an FSU. The CLI provides for this scenario so that a consistent set of commands is available for all types of upgrades.

The command option to perform FSU is applied on the `issu loadversion` command:

```
r1# issu loadversion b stby-disk0:c10k2-p11-mz.2.20040830 forced
```

The addition of the `forced` option disables the automatic rollback following the `issu loadversion` command if the image is unable to return to SSO mode after loading. When the Cisco IOS Software version is incompatible, the highest operating mode will be RPR. The `forced` option allows the system to remain in state "2," as shown in Figure 11, and remain in RPR mode.

Figure 11
Using Forced Option for FSU



Subsequently, when the `issu runversion` command is issued, the system switches to the Cisco IOS Software running on the standby route processor. The switch, which may include resetting and reloading the line cards, affects service, unlike with Cisco IOS ISSU.

Best Practices for Cisco IOS Software ISSU

This section lists some of the best-practice suggestions for consideration in developing operational procedures pertaining to Cisco IOS ISSU. These are guidelines and cautionary statements for your consideration.

- *Avoid manual switchovers.*

  Considering the fact that the Cisco IOS ISSU command structure provides for all the steps necessary to perform an upgrade, any manual switchover commands are disabled once the process is underway and should be avoided. When you begin a Cisco IOS ISSU process by issuing the `issu loadversion` command, either continue the process or end it by issuing the `issu abortversion` command. Make sure the Cisco IOS ISSU *state* has returned to INIT. This is the indication that a Cisco IOS ISSU is not in progress.

- *Avoid card online insertion and removal (OIR).*

  The system behavior may not be deterministic if cards are added or removed from the system while a Cisco IOS ISSU is in progress. Avoid adding or removing cards during an upgrade.

- *Copy Cisco IOS Software prior to Cisco IOS ISSU.*

  The Cisco IOS ISSU process requires that the old and new software be placed on the route processor file systems prior to initiating the Cisco IOS ISSU process. Use standard operating procedures to place the images on the local file systems of the active and standby route processors. When the `issu loadversion` command is issued, internal checks verify that the Cisco IOS Software images included in the command are available on the file systems as specified. If they are not, then the command is rejected.

  Refer to the Cisco IOS Software documentation for information about transferring images to the local file systems.

- *Do not change redundancy mode during the Cisco IOS ISSU process.*

  Refrain from any attempt to alter the redundancy mode during the Cisco IOS ISSU process. Commands to alter the redundancy mode or set the configuration register and boot variables are disabled when the Cisco IOS ISSU process is underway. Remember, Cisco IOS ISSU depends on SSO. If for any reason you want to change the redundancy mode configuration, make sure the Cisco IOS ISSU process is not in progress. Use the `issu abortversion` command if you want to terminate a Cisco IOS ISSU process.

- *MDR and line-card versioning is required.*

  Only Cisco products that support the MDR and line-card versioning or full line-card Cisco IOS ISSU are candidates for Cisco IOS ISSU. Otherwise, service impact, in addition to that of a route processor switchover, will be experienced when line-card software is reset. Always ensure that the line cards have adequate support for Cisco IOS ISSU. It is possible for some line cards to be present that do not support MDR and still perform upgrades. In this case, connections and traffic flowing over these line cards will incur additional service outage. Refer to the appropriate Cisco IOS Software documentation and product release notes for more information.

- *Ensure adequate local file system capacity.*

  When planning for Cisco IOS ISSU, ensure the products have the necessary file system capacity to hold multiple versions of the Cisco IOS Software.

- *Minimize duration of the Cisco IOS ISSU process.*

  The Cisco IOS ISSU process is in progress from the time the `issu loadversion` command is issued until the `issu commitversion` command is issued, unless the `issu abortversion` command is issued. Although the system can theoretically continue in the "Run Version" state after issuing the `issu acceptversion` command indefinitely, you should complete the process within a reasonable amount of time.

You may find some benefit in allowing the system to remain in this state while the system and network connectivity is verified. An extended period even permits an automatic rollback (switchover) to the previous version of Cisco IOS Software if some software defect is encountered while running the new version of software. However, you should make sure that the Cisco IOS ISSU process is completed and the standby route processor brought to the same new version of software as soon as you are comfortable that the new software is operating normally.

- *Use maintenance windows.*

  The support for Cisco IOS ISSU should not suggest that software upgrades be performed without regard to users' service requirements and production network traffic. It remains prudent to plan upgrades according to established and negotiated maintenance windows whenever possible. Cisco IOS ISSU does, however, change the risk assessment criteria for negotiating maintenance windows and minimizes the impact users will see when Cisco IOS Software is upgraded.

- *Do not implement new features while Cisco IOS ISSU is in progress.*

  When the **issu acceptversion** has been issued and the network has been verified, you may be tempted to configure new functions present in the newly active, upgraded software version. The standby route processor is still running the "older" version of software at this point. If you attempt to configure a new feature, the new CLI command will fail to be synchronized to the redundant standby route processor because the back-level version does not have support for the command nor the feature. Therefore, to prevent such a case and eliminate the potential for service disruption if a failure occurred causing a switchover, new CLI commands that fail configuration syntax checking on the standby route processor and configuration synchronization are not permitted. The network administrator sees an error message upon issuing an unsupported command, and the command is rejected. Complete the Cisco IOS ISSU process to allow use of a new feature or function.

- *Disable unsupported features and functions when performing a "downgrade."*

  When there is a need or desire to "downgrade" or begin a Cisco IOS ISSU process to go to an earlier version of Cisco IOS Software, ensure that any new features or functions are disabled prior to starting the process. Any configuration commands that are not available in the "older" target version of software should be removed. After that is done, proceed with the **issu loadversion** and subsequent commands.

Summary

Cisco IOS Software ISSU targets a significant cause of downtime – that caused by planned software upgrades and network system maintenance. Planned downtime represents a significant percentage of total downtime as testified by enterprises and service providers.

Network systems have been proven to increase productivity, and the importance of, and dependence on, continuous access to data, content, applications, and systems will only increase. Global companies and organizations that rely on business partnerships, just-in-time manufacturing, and communication system integration all require nearly 100-percent availability. Any and all downtime is to be avoided if at all possible, and enhancements and upgrades yielding new capabilities and services, changes, and maintenance must go on.

Cisco IOS Software Version 12.2S, first available for the Cisco 10000 in 12.2SB (a 12.2S derivative), now supports Cisco IOS ISSU, which takes advantage of Cisco NSF/SSO and hardware redundancy to permit true in-service software upgrades or version changes while continuously forwarding user traffic. Cisco IOS Software high-availability features combine to lower the impact planned maintenance activities have on network service availability. The result is less downtime and better access to critical systems – anytime, anywhere.

This document has provided information to allow you to take advantage of Cisco IOS ISSU for future upgrades. It examined step-by-step instructions for performing in-service software version changes or maintenance on the Cisco 10000 Series routers. With this knowledge, you can take action to modify your operational procedures to take full advantage of Cisco IOS ISSU and improve network availability.

Cisco 10000 Example

A line-by-line example of the commands to perform Cisco IOS ISSU on the Cisco 10000 is shown in this section. Not all recommended verification steps are shown here. Refer to the "Best Practices for Cisco IOS Software ISSU" section for more detailed information and recommendations regarding operational aspects of Cisco IOS ISSU.

Before beginning, verify the redundancy mode for the system. NSF/SSO should be configured before attempting a Cisco IOS ISSU process.

```
gila#sh redundancy state
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Primary
        Unit ID = 0

Redundancy Mode (Operational) = SSO
Redundancy Mode (Configured)  = SSO
     Split Mode = Disabled
   Manual Swact = Enabled
 Communications = Up

   client count = 31
 client_notification_TMR = 30000 milliseconds
           RF debug mask = 0x0

gila#sh redundancy
Redundant System Information :
------------------------------
       Available system uptime = 9 minutes
Switchovers system experienced = 0
              Standby failures = 0
        Last switchover reason = none

                 Hardware Mode = Duplex
    Configured Redundancy Mode = SSO
     Operating Redundancy Mode = SSO
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot A
        Current Software state = ACTIVE
       Uptime in current state = 9 minutes
                 Image Version = Cisco IOS Software, 10000 Software (C10K2-P11-M), Experimental
Version 12.2(20040825:224856) [wgrupp-c10k_bba_122s_work 102] Copyright (c) 1986-2004 by Cisco
Systems, Inc. Compiled Mon 30-Aug-04 10:29 by wgrupp
                         BOOT = disk0:c10k2-p11-mz.1.20040830,1;
                    CONFIG_FILE =
                        BOOTLDR =
        Configuration register = 0x102
```

```
Peer Processor Information :
-----------------------------
              Standby Location = slot B
        Current Software state = STANDBY HOT
       Uptime in current state = 8 minutes
                 Image Version = Cisco IOS Software, 10000 Software (C10K2-P11-M), Experimental
Version 12.2(20040825:224856) [wgrupp-c10k_bba_122s_work 102] Copyright (c) 1986-2004 by Cisco
Systems, Inc. Compiled Mon 30-Aug-04 10:29 by wgrupp
                           BOOT = disk0:c10k2-p11-mz.1.20040830,1;
                  CONFIG_FILE =
                      BOOTLDR =
        Configuration register = 0x102
```

Notice from the configuration that the system is in SSO mode and that slot A - PRE A is the active route processor and slot B - PRE B is the standby route processor. Both PREs are running the same Cisco IOS Software image, disk0:c10k2-p11-mz.1.20040830.

Next we show the Cisco IOS ISSU state:

```
gila#sh issu state detail
                           Slot = A
                     RP State = Active
                   ISSU State = Init
                Boot Variable = N/A
              Operating Mode = SSO
            Primary Version = N/A
          Secondary Version = N/A
            Current Version = disk0:c10k2-p11-mz.1.20040830

                           Slot = B
                     RP State = Standby
                   ISSU State = Init
                Boot Variable = N/A
              Operating Mode = SSO
            Primary Version = N/A
          Secondary Version = N/A
            Current Version = disk0:c10k2-p11-mz.1.20040830
```

Again, we see similar information.

The new version of the Cisco IOS Software must be present on both route processors. This is evident if we display the directory information for each of the route processors (PREs).

```
gila#dir disk0:
Directory of disk0:/

    1  -rw-    16864340  Jul 16 2004 01:59:42 -04:00  c10k2-p11-mz.122-16.BX1.bin
    2  -rw-     2530912  Jul 16 2004 02:00:04 -04:00  c10k2-eboot-mz.122-16.BX1.bin
3  -rw-    20172208  Aug 30 2004 16:25:56 -04:00  c10k2-p11-mz.1.20040830
    4  -rw-    20171492  Aug 31 2004 12:25:34 -04:00  c10k2-p11-mz.2.20040830
```

```
64253952 bytes total (4509696 bytes free)
gila#dir stby-disk0:
Directory of stby-disk0:/

    1  -rw-    16864340  Jul 16 2004 09:00:26 -04:00  c10k2-p11-mz.122-16.BX1.bin
    2  -rw-     2530912  Jul 16 2004 09:00:46 -04:00  c10k2-eboot-mz.122-16.BX1.bin
    3  -rw-    20172208  Aug 30 2004 16:28:44 -04:00  c10k2-p11-mz.1.20040830
    4  -rw-    20171492  Aug 31 2004 12:30:20 -04:00  c10k2-p11-mz.2.20040830

64253952 bytes total (4509696 bytes free)
```

The current running version of Cisco IOS Software and the new version of the software are available on both route processors. We are ready to initiate the Cisco IOS ISSU process.

```
gila#issu loadversion a disk0:c10k2-p11-mz.2.20040830 b stby-disk0:c10k2-p11-mz.2.20040830
```

A series of messages is logged showing progress of the activity. The standby route processor is loaded with the new Cisco IOS Software version as specified in the command **c10k2-p11-mz.2.20040830**.

When complete, the standby returns to "hot" status.

```
gila#sh issu state
                        Slot = A
                  RP State = Active
                 ISSU State = Load Version
               Boot Variable = disk0:c10k2-p11-mz.1.20040830,1;

                        Slot = B
                  RP State = Standby
                 ISSU State = Load Version
               Boot Variable = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;
gila#
gila#sh red stat
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Primary
        Unit ID = 0

Redundancy Mode (Operational) = SSO
Redundancy Mode (Configured)  = SSO
     Split Mode = Disabled
   Manual Swact = Enabled
 Communications = Up

   client count = 31
 client_notification_TMR = 30000 milliseconds
           RF debug mask = 0x0
```

At this point, the system is ready to switch over and run the new version of Cisco IOS Software that has been loaded on the standby route processor.

```
gila#issu runversion b stby-disk0:c10k2-p11-mz.2.20040830
```

Upon entering this command, a SSO switchover is performed and NSF procedures are invoked if so configured.

When complete, the system runs the new version of software and the previously active route processor (PRE) now becomes the standby route processor. The standby route processor is reset and reloaded, but remains on the previous version of software and comes back online in standby-hot status.

To connect to the newly active route processor and verify conditions:

```
gila#sh red
Redundant System Information :
------------------------------
       Available system uptime = 24 minutes
Switchovers system experienced = 1
              Standby failures = 0
        Last switchover reason = user initiated

                 Hardware Mode = Duplex
    Configured Redundancy Mode = SSO
     Operating Redundancy Mode = SSO
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot B
        Current Software state = ACTIVE
       Uptime in current state = 8 minutes
                 Image Version = Cisco IOS Software, 10000 Software (C10K2-P11-M), Experimental
Version 12.2(20040825:224856) [wgrupp-c10k_bba_122s_work 103] Copyright (c) 1986-2004 by Cisco
Systems, Inc. Compiled Mon 30-Aug-04 11:50 by wgrupp
                         BOOT = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;
                   CONFIG_FILE =
                      BOOTLDR =
        Configuration register = 0x102

Peer Processor Information :
------------------------------
               Standby Location = slot A
        Current Software state = STANDBY HOT
       Uptime in current state = 6 minutes
                 Image Version = Cisco IOS Software, 10000 Software (C10K2-P11-M), Experimental
Version 12.2(20040825:224856) [wgrupp-c10k_bba_122s_work 102] Copyright (c) 1986-2004 by Cisco
Systems, Inc. Compiled Mon 30-Aug-04 10:29 by wgrupp
                         BOOT = disk0:c10k2-p11-mz.1.20040830,1;
                   CONFIG_FILE =
                      BOOTLDR =
        Configuration register = 0x102
```

```
gila#sh issu state
                        Slot = B
                   RP State = Active
                  ISSU State = Run Version
                Boot Variable = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;

                        Slot = A
                   RP State = Standby
                  ISSU State = Run Version
                Boot Variable = disk0:c10k2-p11-mz.1.20040830,1;

gila#sh issu state det
                        Slot = B
                   RP State = Active
                  ISSU State = Run Version
                Boot Variable = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;
              Operating Mode = SSO
            Primary Version = disk0:c10k2-p11-mz.2.20040830
          Secondary Version = disk0:c10k2-p11-mz.1.20040830
            Current Version = disk0:c10k2-p11-mz.2.20040830

                        Slot = A
                   RP State = Standby
                  ISSU State = Run Version
                Boot Variable = disk0:c10k2-p11-mz.1.20040830,1;
              Operating Mode = SSO
            Primary Version = disk0:c10k2-p11-mz.2.20040830
          Secondary Version = disk0:c10k2-p11-mz.1.20040830
            Current Version = disk0:c10k2-p11-mz.1.20040830
```

Notice that the new active route processor (PRE) is now running the new version and the standby route processor (PRE) is running the old version and is in the standby-hot status.

Remember, the Cisco IOS ISSU rollback timer is running, so we must enter the `issu acceptversion` command within the timer period or the Cisco IOS ISSU process terminates and the system reverts back to the previous version by switching over to the standby route processor.

You can see the current timer information using the command `show issu rollback-timer`.

```
gila#show issu rollback-timer
        Rollback Process State = In progress
      Configured Rollback Time = 45:00
       Automatic Rollback Time = 29:03
```

The “automatic rollback time” indicates the amount of time left before an automatic rollback will occur.

```
gila#issu acceptversion b disk0:c10k2-p11-mz.2.20040830

gila#sh issu state
                        Slot = B
                   RP State = Active
```

```
                   ISSU State = Run Version
                Boot Variable = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;

                        Slot = A
                   RP State = Standby
                   ISSU State = Run Version
                Boot Variable = disk0:c10k2-p11-mz.1.20040830,1;

gila#sh issu state det
                        Slot = B
                   RP State = Active
                   ISSU State = Run Version
                Boot Variable = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;
              Operating Mode = SSO
             Primary Version = disk0:c10k2-p11-mz.2.20040830
           Secondary Version = disk0:c10k2-p11-mz.1.20040830
             Current Version = disk0:c10k2-p11-mz.2.20040830

                        Slot = A
                   RP State = Standby
                   ISSU State = Run Version
                Boot Variable = disk0:c10k2-p11-mz.1.20040830,1;
              Operating Mode = SSO
             Primary Version = disk0:c10k2-p11-mz.2.20040830
           Secondary Version = disk0:c10k2-p11-mz.1.20040830
             Current Version = disk0:c10k2-p11-mz.1.20040830

gila#sh red stat
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Secondary
        Unit ID = 1

Redundancy Mode (Operational) = SSO
Redundancy Mode (Configured)  = SSO
     Split Mode = Disabled
   Manual Swact = Enabled
 Communications = Up

   client count = 31
 client_notification_TMR = 30000 milliseconds
           RF debug mask = 0x0
```

Nothing changes after entering the `issu acceptversion` command other than the rollback timer is stopped.

Of course, we could have aborted the Cisco IOS ISSU process at any time by entering the `issu abortversion` command.

At this stage we want to verify all operation and all connectivity.

When completely satisfied that the new version is functioning as expected, we can bring the standby route processor up to the current Cisco IOS Software version using the `issu commitversion` command.

```
gila#issu commitversion a stby-disk0:c10k2-p11-mz.2.20040830
```

The standby route processor is now reset and reloaded with the new Cisco IOS Software version and returned to standby-hot status.

```
gila#sh red state
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Secondary
        Unit ID = 1

Redundancy Mode (Operational) = SSO
Redundancy Mode (Configured)  = SSO
     Split Mode = Disabled
   Manual Swact = Enabled
 Communications = Up

   client count = 31
 client_notification_TMR = 30000 milliseconds
           RF debug mask = 0x0

gila#sh red
Redundant System Information :
------------------------------
       Available system uptime = 35 minutes
Switchovers system experienced = 1
              Standby failures = 1
        Last switchover reason = user initiated

                 Hardware Mode = Duplex
    Configured Redundancy Mode = SSO
     Operating Redundancy Mode = SSO
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot B
        Current Software state = ACTIVE
       Uptime in current state = 18 minutes
                 Image Version = Cisco IOS Software, 10000 Software (C10K2-P11-M), Experimental
Version 12.2(20040825:224856) [wgrupp-c10k_bba_122s_work 103] Copyright (c) 1986-2004 by Cisco
Systems, Inc. Compiled Mon 30-Aug-04 11:50 by wgrupp
                         BOOT = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;
                    CONFIG_FILE =
                        BOOTLDR =
        Configuration register = 0x102
```

```
Peer Processor Information :
-----------------------------
              Standby Location = slot A
        Current Software state = STANDBY HOT
       Uptime in current state = 4 minutes
                 Image Version = Cisco IOS Software, 10000 Software (C10K2-P11-M), Experimental
Version 12.2(20040825:224856) [wgrupp-c10k_bba_122s_work 103] Copyright (c) 1986-2004 by Cisco
Systems, Inc. Compiled Mon 30-Aug-04 11:50 by wgrupp
                          BOOT = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x102

gila#sh issu state
                          Slot = B
                    RP State = Active
                  ISSU State = Init
                 Boot Variable = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;

                          Slot = A
                    RP State = Standby
                  ISSU State = Init
                 Boot Variable = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;

gila#sh issu state det
                          Slot = B
                    RP State = Active
                  ISSU State = Init
                 Boot Variable = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;
                Operating Mode = SSO
              Primary Version = N/A
            Secondary Version = N/A
              Current Version = disk0:c10k2-p11-mz.2.20040830

                          Slot = A
                    RP State = Standby
                  ISSU State = Init
                 Boot Variable = disk0:c10k2-p11-mz.2.20040830,12;disk0:c10k2-p11-mz.1.20040830,1;
                Operating Mode = SSO
              Primary Version = N/A
            Secondary Version = N/A
              Current Version = disk0:c10k2-p11-mz.2.20040830
```

The Cisco IOS ISSU process is complete. At this stage, any further Cisco IOS Software version upgrade or downgrade requires a new Cisco IOS ISSU process.

Appendix

Cisco IOS ISSU Design and State Progression

The Cisco IOS ISSU process is designed using the following global state variables:

- Primary Version (PV)
- Secondary Version (SV)
- ISSU State (IS)
- Current Version (CV)
- Rollback Timer (issuRollbackTimer)

The first four variables in the list are stored in the nonvolatile RAM (NVRAM) and used by the boot program to boot the appropriate image. These variables are updated as the Cisco IOS ISSU CLI commands are issued. The variables are written back into the NVRAM prior to resetting or switching over the route processors. This helps ensure that the updated values are available during the boot process and that the appropriate image is booted.

The possible Cisco IOS ISSU states follow:

- *INIT state* – This means that the Cisco IOS ISSU process is not in progress. In this state, the Primary Version (PV) and the Secondary Version (SV) are not set.
- *LV (Load Version) state* – This means the active route processor runs the old image and the standby route processor runs the new image.
- *RV (Run Version) state* – This means the active route processor runs the new image and the standby route processor runs the old image.
- *LV-SO (Load Version with Switchover)* – This is the state when a switchover takes place during the Load Version state. This state is identical to the Run Version state.
- *RV-SO (Run Version with Switchover)* – This state is arrived at when a switchover takes place during the Run Version state. This state is identical to the Load Version state.
- *SR (System Reset) state* – This is a transitory state; it occurs only if the system (that is, both active and standby route processors) is reset during the Cisco IOS ISSU process; that is, during any one of the Load Version, Load Version with Switchover, Run Version, or Run Version with Switchover states.

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: 31 0 20 357 1000
Fax: 31 0 20 357 1100

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the **Cisco Website at www.cisco.com/go/offices.**

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica
Croatia • Cyprus • Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR
Hungary • India • Indonesia • Ireland • Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico
The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal • Puerto Rico • Romania • Russia
Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden • Switzerland • Taiwan
Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

Printed in the USA C07-339629-00 03/06