Consolidated Packages and SubPackages Management

This document discusses how consolidated packages and software subpackages (individual and optional) are run and managed on the Cisco cBR Series Converged Broadband Router . It contains the following sections:

- Finding Feature Information, on page 1
- Running the Cisco cBR Series Routers Using Individual and Optional SubPackages: An Overview , on page 2
- Running the Cisco cBR Series Routers Using a Consolidated Package: An Overview , on page 2
- Running the Cisco cBR Series Routers: A Summary , on page 3
- Software File Management Using Command Sets , on page 4
- Managing and Configuring the Router to Run Using Consolidated Packages and Individual SubPackages, on page 5
- Upgrading Individual SubPackages, on page 25
- Additional References, on page 33
- Feature Information for Consolidated Packages and SubPackages Management, on page 33

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Running the Cisco cBR Series Routers Using Individual and Optional SubPackages: An Overview

The Cisco cBR Series Converged Broadband Router can be configured to run using individual subpackages and optional subpackages.

When the router is configured to run using individual and optional subpackages:

- Each individual subpackage within a consolidated package is extracted onto the router as its own file.
- Additionally, any optional subpackages must be separately downloaded and stored in the same directory with the provisioning file and the other individual subpackages that have been extracted.
- The router then runs by accessing each file as needed for operational purposes. All individual and optional subpackage files must be stored in the same directory on the router for the router to run properly using individual subpackages.

When the router runs using the individual and optional subpackages, the router needs to be configured to boot using the provisioning file that was included in the consolidated package with the individual subpackage files. This provisioning file must also be in the same directory as the individual and optional subpackage files. The router boots faster when configured to run using individual and optional subpackages than it does when configured to run using a consolidated package.

A Cisco cBR Series Router cannot be configured to run individual and optional subpackages stored on a TFTP or any other network server. To use this method of running the router, copy the individual and optional subpackages along with the provisioning file onto the bootflash: file system.

Running the Cisco cBR Series Routers Using a Consolidated Package: An Overview

The Cisco cBR Series Converged Broadband Router can also be configured to run using a consolidated package.

Note Booting the router from a consolidated package is not supported for installation of optional subpackages.

When the router is configured to run using a consolidated package, the entire consolidated package file is copied onto the router or accessed by the router via TFTP or another network transport method. The router runs using the consolidated package file.

A router configured to run using a consolidated package is booted by booting the consolidated package file. Because this file is large, the boot process for routers running using the consolidated package is slower than the boot process for routers running individual subpackages.

A router configured to run using a consolidated package does have some advantages over a router configured to run individual subpackages. First, a consolidated package can be booted and utilized using TFTP or another network transport method. Secondly, configuring the router to use the one consolidated package file is easier

than managing several individual subpackage files. Running the router using a consolidated package may be the right method of running the router in certain networking environments.

The consolidated package should be stored on bootflash:, usb[0-1]:, or a remote file system when this method is used to run the router.

Running the Cisco cBR Series Routers: A Summary

The advantages of running your router using individual subpackages include:

- The router boots fastest when booted using the individual subpackage boot approach.
- Individual subpackages can be upgraded instead of the complete consolidated image.

The advantages of running your router using a consolidated package include:

- Simplified installation—Only one software file needs to be managed instead of several separate images.
- Storage—A consolidated package, unlike individual subpackages, can be used to run the router while being stored in bootflash:, on a USB Flash disk, or on a network server. A consolidated package can be booted and utilized using TFTP or another network transport method, while the individual subpackage method requires the individual subpackage files to be copied into the bootflash: file directory on the router.

Approach	Advantages	Disadvantages
Individual and optional subpackages **Note** This method is required if you need to install any optional subpackages for your system.	Faster boot time.	• Multiple software subpackages more difficult to manage. • Cannot be booted from TFTP or any other network server. If you are going to use the individual subpackage boot method, each individual subpackage file must be placed in the bootflash: directory. • Individual and optional subpackage files and the provisioning file must be stored in bootflash:.
Consolidated Package	• Easier management. Only have to manage one file instead of many files. • A consolidated package file can be stored in bootflash:, on any TFTP or other network server.	Slower boot times and lessened maximum system scalability because the larger image must be processed at all times.

Software File Management Using Command Sets

Software files can be managed on the Cisco cBR Series Converged Broadband Router using the following distinct command sets.

The request platform Command Set

The **request platform software package** command is part of the larger request platform command set being introduced on the Cisco cBR Series Converged Broadband Router .

The **request platform software package** command, which can be used to upgrade individual subpackages and a complete consolidated package, is used to upgrade software on the Cisco cBR Series Converged Broadband Router . Notably, the **request platform software package** command is the recommended way of performing an individual subpackage upgrade, and also provides the only method of no-downtime upgrades of individual subpackages on the router when the router is running individual subpackages.

The **request platform software package** command requires that the destination device or process be specified in the command line, so the commands can be used to upgrade software on both an active or a standby processor.

Command Syntax

request platform software package install rp *rp-slot-number* **file** *file-URL*

where

- *rp-slot-number* is the number of the RP slot and
- *file-URL* is the path to the file being used to upgrade the Router.

The copy Command

The **copy** command can be used to move consolidated packages and individual subpackages onto the router, though using this command to move individual subpackage files from one storage area to another is often inefficient (in these scenarios, it is almost always preferable to move the consolidated package, then extract the subpackages, or to extract the subpackages without moving the consolidated package).

To upgrade a consolidated package on the Cisco cBR Series Converged Broadband Router , copy the consolidated package onto a file system, usually bootflash: , using the **copy** command as you would on most other Cisco routers. After making this copy, configure the router to boot using the consolidated package file.

To upgrade the router and reboot using individual subpackages, copy the consolidated package onto the router using the **copy** command, enter the **request platform software package expand** command to extract the individual subpackages, and configure the router to boot using subpackages. Other methods, such as copying each individual subpackage in the same consolidated package from a directory or using the request platform software package command to extract the subpackages onto a router directory are also usable, though copying individual subpackages is often inefficient.

Managing and Configuring the Router to Run Using Consolidated Packages and Individual SubPackages

This section describes methods that are used to manage and configure the packages, sub-packages and patches on the router.

Cable Line Card Process Restart

The last step, in many of the methods used to upgrade and manage the packages and sub-packages, is to reload the router or the component for which the software is being configured or upgraded. Reloading the router or the components involves the following disadvantages:

- service disruption (limited)
- loss of modem configuration data
- time consumption in rebooting the line cards and other components

The N+1 line card high availability (LCHA) system reboots the active and the standby line cards whenever a package is upgraded on a line card. This occurs for sub-package upgrade for Field Replaceable Units (FRUs) on the line card as well. Every time an upgrade is done to a package or sub-package, the line card must be rebooted. The time taken for the package upgrade on N number of active line cards, the total number of reboots would be 2xN. This is time-consuming and may affect services on the rebooting line cards.

To avoid the disadvantages of reloading the router and the line cards, use the Cable Line Card Process Restart features when you upgrade packages on the RF line cards.

Note Do not use the process restart features without upgrading or installing packages on the RF line cards.

Primarily there are two features that allow you to restart Cable Line Card processes:

- Cable Line Card Control Plane Process Restart
- Cable Line Card Upstream Scheduler Process Restart

Cable Line Card Control Plane Process Restart

The Cable Line Card Control Plane Process Restart feature provides the following advantages:

- Simplified package upgrade without LCHA based reboot of active and standby line cards.
- Restart of specific processes without service disruption.
- All the modem configuration data is recovered after the IOSd process restarts.
- Changes in the modem configuration data during the IOSd process restart is reconciled after the IOSd process is restarted.
- The secondary line card shutdown and un-shutdown processes occur automatically.

Restrictions:

- To upgrade a line card IOS using the Cable Line Card Control Plane Process Restart feature, the sub-package must have the patches for cbrsup-clcios and cbrsup-clciosdb.
- IOSd and us-scheduler restart are supported. You can restart IOSd only, which requires IOSd/IOSdb packages or restart the us-scheduler only, which requires clc-docsis package. You can also restart both using a single command, where the us-scheduler is restarted first and then the IOSd is restarted. In this case, IOSd/IOSdb and clc-docsis packages are required.

Other restrictions include the following:

- The IOSd process can be restarted only on the primary active RF line cards.
- Any secondary RF line card must be shut down before restarting the IOSd process.
- The IOSd process restart does not work when double failures (i.e. termination of IOSd and one or more process at the same time) occur. The double failures result in line card reload.
- The restart of the next IOSd (i.e. IOSd process on the next RF line card) does not occur until the current IOSd process recovers fully.

The Cable Line Card Control Plane Process Restart feature provides the following restart options:

- Restart a specific slot using the **request platform software process restart** command with the **slot** *slot number* option.
- Restart all the line cards without specifying a specific slot, using the **request platform software process restart** command without the **slot** *slot number* option.
- Interval based restart option of all line card using the **request platform software process restart** command with the **interval** *secs* option.

Important points to remember:

- Before using the Cable Line Card Control Plane Process Restart feature, shut down the secondary line card, using the **hw-module slot shutdown** command.
- When the Cable Line Card Control Plane Process Restart feature is started, the secondary line card shuts down automatically.
- Before using the Cable Line Card Control Plane Process Restart feature, shut down the secondary line card, using the **hw-module slot shutdown** command.
- When the Cable Line Card Control Plane Process Restart feature is started, the secondary line card shuts down automatically.
- To prevent reloading the line card after sub-package upgrade, use the **request platform software package install node file** command with the **noreload linecard** option.
- Verify the restart process using the **show platform software ios slot** *slot number* **restart info** command.

Restart on Crash

The cable line card control plane and upstream scheduler process restarts automatically after a crash. After restarting process, secondary line card is reset.

Restrictions:

- Restart on crash supported only on primary active line cards

The table below lists the cable line card behaviors when crash happens under different rules.

Table 1: Cable Line Card Process Restart Policy Matrix

	Secondary Cable Line Card Present	Crash	Secondary Cable Line Card state
LCHA Preferred	Yes	Cable line card reset	Active after switchover
Process restart Enabled	No	Restart	N/A
LCHA Preferred	Yes	Cable line card reset	Active after switchover
Process restart Disabled	No	Cable line card reset	N/A
No LCHA Preferred	Yes	Process restart	Secondary cable line card resets after process restart on primary cable line card
Process restart Enabled	No	Process restart	N/A
No LCHA Preferred	Yes	Cable line card reset	Active after switchover
Process restart Disabled	No	Cable line card reset	N/A

Note

- Manual process restart does not depend on policy configured, and will shut/unshut secondary cable line card if present.
- "LCHA preferred" and "process restart enable" are default. User can set these two parameters using disable-auto-restart and lcha-preferred commands. For the details, see http://www.cisco.com/c/en/us/td/docs/cable/cmts/cmd_ref/b_cmts_cable_cmd_ref.html.
- Behavior is the same for Control Plane Process Restart and Upstream Scheduler Process Restart.
- Secondary cable line card in standby mode is considered present.

The restart retry limit feature is added to the Cable Line Card Process Restart, it is applicable only to restart on crash. Using this feature, the customer can set a restart retry time limit, if the process cannot restart successfully within this limit, the line card will reload. This feature can prevent the line card from continuous restart when restart failed.

Using the Cable Line Card Control Plane Process Restart Feature

To use the Cable Line Card Control Plane Process Restart Feature, install the RF line card sub-package upgrade using the command.

Procedure

Step 1 Install the RF line card sub-package upgrade using the **request platform software package install node file noreload linecard** command.

```
Router#request platform software package install node file bootflash:sp/cbr_patch.9.0.tar
noreload linecard
```

Step 2 Use the **request platform software process restart** command to restart the RF line card IOSd process on all the cable line cards sequentially.

```
Router# request platform software process restart
```

Step 3 Use the **request platform software process restart slot** *slot#* command to restart the RF line card IOSd process on a specific cable line card.

```
Router# request platform software process restart slot 7
```

Configuring the Cable Line Card Control Plane Process Restart Retry Limit

To configure the Cable Line Card Control Plane Process Restart Retry Limit, complete the following procedure:

```
enable
configure terminal
process-restart
lc-control-plane-timeout time
restart-retry retry-times
exit
```

Examples for Cable Line Card Control Plane Process Restart Feature

This section provides the sample outputs for the commands used in the Cable Line Card Control Plane Process.

This example shows the output of the **request platform software package install node file** command with the **noreload linecard** option that installs the sub-package upgrade:

```
Router# request platform software package install node file bootflash:sp/cbr_patch.9.0.tar
noreload linecard
Image file expanded and copied
Expanding image file: stby-bootflash:sp/cbr_patch.9.0.tar
Image file expanded and copied
Finished image file expansion
Found clc package
STAGE 1: Installing software on standby RP
=========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0
--- Starting installation state synchronization ---
Finished installation state synchronization
```

```
--- Starting local lock acquisition on R1 ---
Finished local lock acquisition on R1
--- Starting file path checking ---
Finished file path checking
--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
Found cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
Found cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification
--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction
--- Starting ISSU compatibility verification ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatibility verification
--- Starting list of software package changes ---
Old files list:
Removed cbrsup-clcios.2015-03-23_17.28_haolin2.SSA.pkg
Removed cbrsup-clciosdb.2015-03-23_17.28_haolin2.SSA.pkg
New files list:
Added cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
Added cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
Finished list of software package changes
--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
Committing provisioning file
Finished commit of software changes
--- Starting analysis of software changes ---
Finished analysis of software changes
--- Starting update running software ---
Blocking peer synchronization of operating information
Creating the command set placeholder directory
Finding latest command set
Finding latest command shortlist lookup file
Finding latest command shortlist file
Assembling CLI output libraries
Assembling CLI input libraries
Assembling Dynamic configuration files
Applying interim IPC and database definitions
Replacing running software
Replacing CLI software
Restarting software
Restarting software: target frus filtered out ... skipped
Applying final IPC and database definitions
Generating software version information
Notifying running software of updates
Unblocking peer synchronization of operating information
Unmounting old packages
```

```
Cleaning temporary installation files
Finished update running software
SUCCESS: Finished installing software.
STAGE 2: Installing software on active RP
=========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0
--- Starting installation state synchronization ---
Finished installation state synchronization
--- Starting file path checking ---
Finished file path checking
--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
Found cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
Found cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification
--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction
--- Starting ISSU compatibility verification ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatibility verification
--- Starting impact testing ---
Checking operational impact of change
Finished impact testing
--- Starting list of software package changes ---
Old files list:
Removed cbrsup-clcios.2015-03-23_17.28_haolin2.SSA.pkg
Removed cbrsup-clciosdb.2015-03-23_17.28_haolin2.SSA.pkg
New files list:
Added cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
Added cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
Finished list of software package changes
--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
Committing provisioning file
Finished commit of software changes
--- Starting analysis of software changes ---
Finished analysis of software changes
--- Starting update running software ---
Blocking peer synchronization of operating information
Creating the command set placeholder directory
Finding latest command set
Finding latest command shortlist lookup file
Finding latest command shortlist file
Assembling CLI output libraries
Assembling CLI input libraries
Assembling Dynamic configuration files
```

```
Applying interim IPC and database definitions
Replacing running software
Replacing CLI software
Restarting software
Restarting software: target frus filtered out ... skipped
Applying final IPC and database definitions
Generating software version information
Notifying running software of updates
Unblocking peer synchronization of operating information
Unmounting old packages
Cleaning temporary installation files
Finished update running software
SUCCESS: Finished installing software.
Found clc package
SUCCESS: node ISSU finished successfully.
Invoking cleanup routine
```

This example shows the output of the **show platform software ios restart info** command:

```
Router#show platform software ios 6 restart info
IOSD process restart info:
Process restartable: Yes
IOSD restart state : ACTIVE
Total Modem Count : 31
Active Modem Count : 31
```

This example shows the output of the **request platform software process restart** command:

```
Router# request platform software process restart
--- Upgrading/Restarting LineCard-2 Packages/Processes ---
NOTICE: No upgrades available.
Provide process name in cli if you wish to restart a process
--- Upgrading/Restarting LineCard-3 Packages/Processes ---
Available upgrades
cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
--- Checking for ready state before IOSD upgrade on LineCard-3
Updating Package cbrsup-clcios.2015-03-23_17.28_haolin2.SSA.pkg
---> cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
Restarting ubrclc_k9lc_ms
--- Checking for ready state before IOSDB upgrade on LineCard-3
Updating Package cbrsup-clciosdb.2015-03-23_17.28_haolin2.SSA.pkg
---> cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
Restarting iosdb
SUCCESS: Finished
```

```
Router#show platform software patch 2 info
cbrsup-clciosdb: 3.17.0 (3.0)
cbrsup-clc-firmware: 3.17.0 (0.0)
cbrsup-clcvideo: 3.17.0 (0.0)
cbrsup-clcios: 3.17.0 (3.0)
cbrsup-clccontrol: 3.17.0 (0.0)
cbrsup-clcdocsis: 3.17.0 (0.0)
```

Cable Line Card Upstream Scheduler Process Restart

The Cable Line Card Upstream Scheduler Process Restart feature is used to restart the upstream scheduler process on the RF line cards.

The Cable Line Card Upstream Scheduler Process Restart feature provides the following advantages:

- Allows to restart line card US- Scheduler (CDMAN) process with minimum impact to upstream traffic and no impact to downstream traffic.
- All modem configuration data recovered to the state before restart.
- Changes in modem configuration data during restart are reconciled.
- New modems coming online are blocked during restart.
- The **request platform software package restart** command is used to upgrade the new DOCSIS sub-pkg patch without reloading the line card.
- Effective with Cisco IOS-XE Release 3.18.0S, the card restarts automatically after a crash. For more information, see Restart on Crash, on page 6 section.

Restrictions:

The following restrictions apply to the :

- The Upstream Scheduler process can be restarted only on the primary active RF line cards.
- The Upstream Scheduler process restart does not work when double failures (i.e. termination of Upstream Scheduler and one or more process at the same time) occur. The double failures result in line card reload.
- The restart of the next Upstream Scheduler (i.e. Upstream Scheduler process on the next RF line card) does not occur until the current Upstream Scheduler process recovers fully.

Important points to remember:

- With Cisco IOS-XE Release 3.17.0S, when the Cable Line Card Upstream Scheduler Process Restart feature is started, the secondary line card shuts down automatically.
- To prevent reloading the line card after sub-package upgrade, use the **request platform software package install node file** command with the **noreload linecard** option.
- Verify the restart process using the **show platform software us-scheduler restart info** command.

```
Router# show platform software us-scheduler 3 restart info
us-scheduler process restart info:
Process restartable : Yes
us-scheduler state : RESTART_OPERATIONAL
Features bit map : 0x001e
us-scheduler restart count : 4
```

Using the Cable Line Card Upstream Scheduler Process Restart Feature

To use the Cable Line Card Upstream Scheduler Process Restart Feature, install the RF line card sub-package upgrade using the command.

Procedure

Step 1 Install the RF line card sub-package upgrade using the **request platform software package install node file noreload linecard** command.

```
Router#request platform software package install node file bootflash:sp/cbr_patch.9.0.tar
noreload linecard
```

Step 2 Use the **request platform software process restart** command to restart the Upstream Scheduler process on all the cable line cards sequentially.

```
Router# request platform software process restart
```

Step 3 Use the **request platform software process restart slot** *slot#* command to restart the Upstream Scheduler process on a specific cable line card.

```
Router# request platform software process restart slot 7
```

Configuring the Cable Line Card Control Plane Process Restart Retry Limit

To configure the Cable Line Card Control Plane Process Restart Retry Limit, complete the following procedure:

```
enable
configure terminal
process-restart
lc-us-scheduler-timeout time
restart-retry retry-times
exit
```

Examples for Cable Line Card Upstream Scheduler Process Restart Feature

This section provides the sample outputs for the commands used in the Cable Line Card Upstream Scheduler Process Restart Feature.

This example shows the output of the **show platform software us-scheduler restart info** command.

```
Router#show platform software us-scheduler 6 restart info
us-scheduler process restart info:
Process restartable : Yes
us-scheduler state : RESTART_OPERATIONAL
Features bit map : 0x001e
us-scheduler restart count : 1
```

This example shows the output when the upstream scheduler process is restarted.

```
Router# request platform software package install node file
```

```
bootflash:subpkg/cbr_patch-3.17.0-patch2.tar noreload linecard
NOTE: Currently node has booted from a provisioning file
NOTE: Going to start a dual rp sub-packages node ISSU install
--- Starting initial file path checking --- Copying
bootflash:subpkg/cbr_patch-3.17.0-patch2.tar to
stby-bootflash:subpkg/cbr_patch-3.17.0-patch2.tar
Finished initial file path checking
--- Starting config-register verification --- Finished config-register verification
--- Starting Checking noreload options --- Finished Checking noreload options
--- Starting image file expansion ---
Expanding image file: bootflash:subpkg/cbr_patch-3.17.0-patch2.tar
Image file expanded and copied
Expanding image file: stby-bootflash:subpkg/cbr_patch-3.17.0-patch2.tar
Image file expanded and copied
Finished image file expansion
Found clc package
STAGE 1: Installing software on standby RP ========================================
--- Starting local lock acquisition on R0 --- Finished local lock acquisition on R0
--- Starting installation state synchronization --- Finished installation state
synchronization
--- Starting local lock acquisition on R1 --- Finished local lock acquisition on R1
--- Starting file path checking ---
Finished file path checking
--- Starting image file verification --- Checking image file names Locating image files and
validating name syntax
Found cbrsup-clcdocsis.2015-10-08_18.10_haolin2.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification
--- Starting candidate package set construction --- Verifying existing software set Processing
candidate provisioning file Constructing working set for
candidate package set Constructing working set for running package set Checking command
output Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete Finished candidate package
set construction
--- Starting ISSU compatibility verification ---
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
Verifying image type compatibility
Checking IPC compatibility with running software Checking candidate package set infrastructure
compatibility Checking infrastructure compatibility with running
software Checking package specific compatibility Finished ISSU compatibility verification
--- Starting list of software package changes --- Old files list:
Removed cbrsup-clcdocsis.BLD_MCP_DEV_LATEST_20151006_133623.SSA.pkg
New files list:
Added cbrsup-clcdocsis.2015-10-08_18.10_haolin2.SSA.pkg
Finished list of software package changes
--- Starting commit of software changes --- Updating provisioning rollback files Creating
```

```
pending provisioning file Committing provisioning file Finished
commit of software changes
--- Starting analysis of software changes --- Finished analysis of software changes
--- Starting update running software --- Blocking peer synchronization of operating
information Creating the command set placeholder directory
Finding latest command set
Finding latest command shortlist lookup file
Finding latest command shortlist file
Assembling CLI output libraries
Assembling CLI input libraries
Assembling Dynamic configuration files
Applying interim IPC and database definitions
Replacing running software
Replacing CLI software
Restarting software
Restarting software: target frus filtered out ... skipped
Applying final IPC and database definitions
Generating software version information
Notifying running software of updates
Unblocking peer synchronization of operating information Unmounting old packages Cleaning
temporary installation files
Finished update running software
SUCCESS: Finished installing software.
STAGE 2: Installing software on active RP ========================================
--- Starting local lock acquisition on R0 --- Finished local lock acquisition on R0
--- Starting installation state synchronization --- Finished installation state
synchronization
--- Starting file path checking ---
Finished file path checking
--- Starting image file verification --- Checking image file names Locating image files and
validating name syntax
Found cbrsup-clcdocsis.2015-10-08_18.10_haolin2.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification
--- Starting candidate package set construction --- Verifying existing software set Processing
candidate provisioning file Constructing working set for candidate
package set Constructing working set for running package set Checking command output
Constructing merge of running and candidate packages Checking if resulting
candidate package set would be complete Finished candidate package set construction
--- Starting ISSU compatibility verification ---
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
Verifying image type compatibility
Checking IPC compatibility with running software Checking candidate package set infrastructure
compatibility Checking infrastructure compatibility with running
software Checking package specific compatibility Finished ISSU compatibility verification
--- Starting impact testing ---
```

```
Checking operational impact of change
Finished impact testing
--- Starting list of software package changes --- Old files list:
Removed cbrsup-clcdocsis.BLD_MCP_DEV_LATEST_20151006_133623.SSA.pkg
New files list:
Added cbrsup-clcdocsis.2015-10-08_18.10_haolin2.SSA.pkg
Finished list of software package changes
--- Starting commit of software changes --- Updating provisioning rollback files Creating
pending provisioning file Committing provisioning file Finished
commit of software changes
--- Starting analysis of software changes --- Finished analysis of software changes
--- Starting update running software --- Blocking peer synchronization of operating
information Creating the command set placeholder directory
Finding latest command set
Finding latest command shortlist lookup file
Finding latest command shortlist file
Assembling CLI output libraries
Assembling CLI input libraries
Assembling Dynamic configuration files
Applying interim IPC and database definitions
Replacing running software
Replacing CLI software
Restarting software
Restarting software: target frus filtered out ... skipped
Applying final IPC and database definitions
Generating software version information
Notifying running software of updates
Unblocking peer synchronization of operating information Unmounting old packages Cleaning
temporary installation files
Finished update running software
SUCCESS: Finished installing software.
Found clc package
SUCCESS: node ISSU finished successfully.
Invoking cleanup routine
Router#

Router#show platform software us-scheduler 2 restart info
us-scheduler process restart info:
Process restartable : Yes
us-scheduler state : RESTART_OPERATIONAL
Features bit map : 0x001e
us-scheduler restart count : 1

Router#request platform software process restart slot 2
--- Upgrading/Restarting LineCard-2 Packages/Processes ---
Available upgrades
cbrsup-clcdocsis.BLD_MCP_DEV_LATEST_20151006_13362
Updating Package cbrsup-clcdocsis.BLD_MCP_DEV_LATEST_20151006_13362
---> cbrsup-clcdocsis.2015-10-08_18.10_haolin2.SSA.pkg
Restarting us-scheduler
SUCCESS: Finished upgrading the LineCard-2

Router#show platform software patch 2 info
cbrsup-clciosdb: 3.17.0 (0.0)
cbrsup-clc-firmware: 3.17.0 (0.0)
cbrsup-clcvideo: 3.17.0 (0.0)
cbrsup-clcios: 3.17.0 (0.0)
cbrsup-clccontrol: 3.17.0 (0.0)
cbrsup-clcdocsis: 3.17.0 (2.0)
```

Note If the upgrade package includes both IOSD-CLC and US-scheduler sub-packages, the **request platform software process restart** command first restarts the Cable Line Card Upstream Scheduler process and then the Cable Line Card Control Plane process.

This example shows the output of the **request platform software process restart** command when the Control Plane and the upstream scheduler process are restarted:

Note If the **slot** keyword is not used, the upstream scheduler process on all the line cards are restarted sequentially.

```
Router#request platform software process restart


--- Upgrading/Restarting LineCard-1 Packages/Processes ---

    Available upgrades
        cbrsup-clcdocsis.2015-09-24_03.09_johuynh.SSA.pkg
        cbrsup-clcios.2015-09-24_03.09_johuynh.SSA.pkg
        cbrsup-clciosdb.2015-09-24_03.09_johuynh.SSA.pkg

    Updating Package  cbrsup-clcdocsis.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clcdocsis.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting us-scheduler
--- Checking for ready state before IOSD upgrade on LineCard-1
    Updating Package  cbrsup-clcios.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clcios.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting ubrclc_k9lc_ms
--- Checking for ready state before IOSDB upgrade on LineCard-1
    Updating Package  cbrsup-clciosdb.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clciosdb.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting iosdb

    SUCCESS: Finished upgrading the LineCard-1


--- Upgrading/Restarting LineCard-2 Packages/Processes ---

    Available upgrades
        cbrsup-clcdocsis.2015-09-24_03.09_johuynh.SSA.pkg
        cbrsup-clcios.2015-09-24_03.09_johuynh.SSA.pkg
        cbrsup-clciosdb.2015-09-24_03.09_johuynh.SSA.pkg

    Updating Package  cbrsup-clcdocsis.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clcdocsis.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting us-scheduler
--- Checking for ready state before IOSD upgrade on LineCard-2
    Updating Package  cbrsup-clcios.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clcios.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting ubrclc_k9lc_ms
--- Checking for ready state before IOSDB upgrade on LineCard-2
    Updating Package  cbrsup-clciosdb.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clciosdb.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting iosdb

    SUCCESS: Finished upgrading the LineCard-2
Router#
```

You can specify a time interval in seconds using the **interval** keyword, between the restarting of two line card processes in the sequence. The default interval is five seconds.

This example shows the configuration of an interval of six seconds.

```
Router#request platform software process restart interval 6
```

Quick Start Software Upgrade

The following instructions provide a quick start version of upgrading the software running the Cisco cBR Series Converged Broadband Router . These instructions assume you have access to the consolidated package and that the files will be stored in a bootflash: file system that is not storing any previously installed subpackages or consolidated packages and that has enough room for the file or files.

Procedure

Step 1 Copy the consolidated package into bootflash: using the **copy** *URL-to-image* **bootflash:** command.

Step 2 If you want to run the router using individual subpackages, enter the **request platform software package expand file bootflash:**/*sub_dir*/*base_image* command. If you want to run the router using a consolidated package, skip this step.

Step 3 Enter the **dir bootflash:** command to verify your consolidated package or your extracted subpackages are in the directory.

Step 4 If you are trying to run individual subpackages, use the **delete bootflash:***base_image* to delete the consolidated package. If you want to run the router using the consolidated package, skip this step.

Step 5 Set up the boot parameters for your boot. Set the configuration register to 0x2 by entering the **config-register 0x2102** global configuration command, and enter the **boot system flash bootflash:***base_image* (if running using the consolidated package) or **boot system flash bootflash:***provisionging-file-name* (if running using individual subpackages) global configuration command.

Step 6 Enter **copy running-config startup-config** to save your configuration.

Step 7 Enter the **reload** command to reload the router and finish the boot. The upgraded software should be running when the reload completes.

Managing and Configuring a Consolidated Package Using the copy Command

To upgrade a consolidated package on the Cisco cBR Series Routers using the **copy** command, copy the consolidated package into the bootflash: directory on the router using the **copy** command as you would on most other Cisco routers. After making this copy, configure the router to boot using the consolidated package file.

In the following example, the consolidated package file is copied onto the bootflash: file system from TFTP. The config-register is then set to boot using **boot system** commands, and the **boot system** commands instruct the router to boot using the consolidated package stored in the bootflash: file system. The new configuration

is then saved using the **copy running-config startup-config** command, and the system is then reloaded to complete the process.

```
Router# dir bootflash:
Directory of bootflash:/

11 drwx 16384 Dec 4 2007 04:32:46 -08:00 lost+found
86401 drwx 4096 Dec 4 2007 06:06:24 -08:00 .ssh
14401 drwx 4096 Dec 4 2007 06:06:36 -08:00 .rollback_timer
28801 drwx 4096 Mar 18 2008 17:31:17 -07:00 .prst_sync
43201 drwx 4096 Dec 4 2007 04:34:45 -08:00 .installer
13 -rw- 45977 Apr 9 2008 16:48:46 -07:00 target_support_output.tgz.tgz

928862208 bytes total (712273920 bytes free)

Router# copy tftp bootflash:
Address or name of remote host []? 172.17.16.81
Source filename []?
/auto/tftp-users/user/cbrsup-universal*.bin Destination filename [cbrsup-universal*.bin]?
Accessing
tftp://172.17.16.81//auto/tftp-users/user/ cbrsup-universal*.bin...

Loading /auto/tftp-users/user/cbrsup-universal*.bin from
172.17.16.81 (via GigabitEthernet0):
!!!
```

```
[OK]
Router# reload
```

Managing and Configuring a Router to Run Using Individual SubPackages From a Consolidated Package

To run the router using individual subpackages from a consolidated package, follow one of the following procedures:

Extracting a Consolidated Package and Booting Using the Provisioning File

T o extract a consolidated package and to boot using provisioning file, perform the following steps:

Procedure

Step 1 Perform one of the following tasks:

a) Copy the consolidated package file (or, in cases where you have every individual subpackage and a provisioning file for the subpackages available, each individual subpackage and the provisioning file) onto the bootflash: file system using the **copy** command. Make sure to copy the consolidated package into the bootflash: file system and directory where you want to store the provisioning file and the individual image subpackages. Enter the **request platform software package expand file bootflash:***url-to-Cisco-IOS-XE-imagename* command with no other option to extract the provisioning file and the individual subpackages out of the consolidated package file and into the current directory in bootflash:. We recommend that you extract the sub packages to an empty directory to simplify the management of these files.

b) Copy the consolidated package file onto any file system on your router, then enter the **request platform software package expand file file-system:***url-to-Cisco-IOS-XE-imagename* **to bootflash:** command to extract the provisioning file and the individual image subpackages onto the bootflash: file system.

Note
After performing this step, do not move any of the files. The bootup process cannot function properly unless all of the subpackages and the provisioning file are located in the same directory. Also, do not rename the subpackage files. Only the provisioning file can be renamed, and the renaming of the provisioning file, if desired, should be done at this step before the router is rebooted.

Step 2 Configure the router to boot using the provisioning file.

The sequence below provides an example that would boot the router using the provisioning file named packages.conf that was stored with the other subpackages in the bootflash: file system:

```
Router(config)# no boot system
Router(config)# config-register 0x2102
Router(config)# boot system flash bootflash:packages.conf
Router(config)# exit
*May 11 01:31:04.815: %SYS-5-CONFIG_I: Configured from console by con
Router# copy running-config startup-config
Building configuration... [OK]
Router# reload
```

Extracting the SubPackages and the Provisioning File: Example 1

The following example shows how to extract the individual subpackages and the provisioning file from a consolidated package that has already been placed in the directory where you want to store the individual subpackages and the provisioning file. We recommend that you extract the sub packages to an empty directory to simplify the management of these files.

Output of the directory before and after the extraction is given to confirm the files were extracted.

```
Router# dir bootflash:
Directory of bootflash:/

11 drwx 16384 Dec 4 2007 04:32:46 -08:00 lost+found
86401 drwx 4096 Dec 4 2007 06:06:24 -08:00 .ssh
14401 drwx 4096 Dec 4 2007 06:06:36 -08:00 .rollback_timer
28801 drwx 4096 Mar 18 2008 17:31:17 -07:00 .prst_sync
43201 drwx 4096 Dec 4 2007 04:34:45 -08:00 .installer
12 -rw- 208904396 May 9 2008 14:36:31 -07:00
cbrsup-universal*.bin
13 -rw- 45977 Apr 9 2008 16:48:46 -07:00 target_support_output.tgz.tgz

928862208 bytes total (503156736 bytes free)

Router# request platform software package expand file bootflash:cbrsup-universal*.bin
Verifying parameters
Validating package type
Copying package files
SUCCESS: Finished expanding all-in-one software package.

Router# dir bootflash:
Directory of bootflash:/

11 drwx 16384 Dec 4 2007 04:32:46 -08:00 lost+found
86401 drwx 4096 Dec 4 2007 06:06:24 -08:00 .ssh
14401 drwx 4096 Dec 4 2007 06:06:36 -08:00 .rollback_timer
28801 drwx 4096 Mar 18 2008 17:31:17 -07:00 .prst_sync
43201 drwx 4096 Dec 4 2007 04:34:45 -08:00 .installer
12 -rw- 208904396 May 9 2008 14:36:31 -07:00
cbrsup-universal*.bin
57611 -rw- 47071436 May 22 2008 11:26:23 -07:00 cbr000rp1-espbase.02.01.00.122-33.XNA.pkg
57602 -rw- 5740 May 22 2008 11:26:22 -07:00
cbr000rp1-packages-adventerprisek9.02.01.00.122-33.XNA.conf
57612 -rw- 20334796 May 22 2008 11:26:24 -07:00
cbr000rp1-rpaccess.02.01.00.122-33.XNA.pkg
57613 -rw- 22294732 May 22 2008 11:26:24 -07:00 cbr000rp1-rpbase.02.01.00.122-33.XNA.pkg
57614 -rw- 21946572 May 22 2008 11:26:25 -07:00 cbr000rp1-rpcontrol.02.01.00.122-33.XNA.pkg
57615 -rw- 48099532 May 22 2008 11:26:26 -07:00
cbr000rp1-rpios-adventerprisek9.02.01.00.122-33.XNA.pkg
57616 -rw- 34324684 May 22 2008 11:26:27 -07:00 cbr000rp1-sipbase.02.01.00.122-33.XNA.pkg
57617 -rw- 22124748 May 22 2008 11:26:28 -07:00 cbr000rp1-sipspa.02.01.00.122-33.XNA.pkg
57603 -rw- 6256 May 22 2008 11:26:28 -07:00 packages.conf
13 -rw- 45977 Apr 9 2008 16:48:46 -07:00 target_support_output.tgz.tgz

928862208 bytes total (286662656 bytes free)
```

Extracting the SubPackages, Configuring the Router to Boot Using the Provisioning File, and Reloading the Router: Example 2

In the following example, the provisioning file and the individual subpackages are extracted from a consolidated package. The router is then configured to boot using the provisioning file. This example also shows the

config-register being set and the running configuration being saved because these tasks must be performed for the router to reload properly. The router is then reloaded to complete the process.

```
Router# dir bootflash:
Directory of bootflash:/

11 drwx 16384 Dec 4 2007 04:32:46 -08:00 lost+found
86401 drwx 4096 Dec 4 2007 06:06:24 -08:00 .ssh
14401 drwx 4096 Dec 4 2007 06:06:36 -08:00 .rollback_timer
28801 drwx 4096 Mar 18 2008 17:31:17 -07:00 .prst_sync
43201 drwx 4096 Dec 4 2007 04:34:45 -08:00 .installer
12 -rw- 208904396 May 9 2008 14:36:31 -07:00
cbrsup-universal*.bin
13 -rw- 45977 Apr 9 2008 16:48:46 -07:00 target_support_output.tgz.tgz

928862208 bytes total (503156736 bytes free)

Router# request platform software package expand file bootflash:cbrsup-universal*.bin
Verifying parameters
Validating package type
Copying package files
SUCCESS: Finished expanding all-in-one software package.

Router# dir bootflash:
Directory of bootflash:/

11 drwx 16384 Dec 4 2007 04:32:46 -08:00 lost+found
86401 drwx 4096 Dec 4 2007 06:06:24 -08:00 .ssh
14401 drwx 4096 Dec 4 2007 06:06:36 -08:00 .rollback_timer
28801 drwx 4096 Mar 18 2008 17:31:17 -07:00 .prst_sync
43201 drwx 4096 Dec 4 2007 04:34:45 -08:00 .installer
12 -rw- 208904396 May 9 2008 14:36:31 -07:00
cbrsup-universal*.bin
57611 -rw- 47071436 May 22 2008 11:26:23 -07:00 cbr000rp1-espbase.02.01.00.122-33.XNA.pkg
57602 -rw- 5740 May 22 2008 11:26:22 -07:00
cbr000rp1-packages-adventerprisek9.02.01.00.122-33.XNA.conf
57612 -rw- 20334796 May 22 2008 11:26:24 -07:00 cbr000rp1-rpaccess.02.01.00.122-33.XNA.pkg
57613 -rw- 22294732 May 22 2008 11:26:24 -07:00 cbr000rp1-rpbase.02.01.00.122-33.XNA.pkg
57614 -rw- 21946572 May 22 2008 11:26:25 -07:00
cbr000rp1-rpcontrol.02.01.00.122-33.XNA.pkg
57615 -rw- 48099532 May 22 2008 11:26:26 -07:00
cbr000rp1-rpios-adventerprisek9.02.01.00.122-33.XNA.pkg
57616 -rw- 34324684 May 22 2008 11:26:27 -07:00 cbr000rp1-sipbase.02.01.00.122-33.XNA.pkg
57617 -rw- 22124748 May 22 2008 11:26:28 -07:00
cbr000rp1-sipspa.02.01.00.122-33.XNA.pkg
57603 -rw- 6256 May 22 2008 11:26:28 -07:00 packages.conf
13 -rw- 45977 Apr 9 2008 16:48:46 -07:00 target_support_output.tgz.tgz

928862208 bytes total (286662656 bytes free)

Router(config)# no boot system
Router(config)# config-register 0x2102
Router(config)# boot system flash bootflash:packages.conf
Router(config)# exit
Router# copy run start
Router# reload
```

Copying a Set of Individual SubPackage Files, and Booting Using a Provisioning File

To copy a set of individual subpackage files and to boot using a provisioning file, perform the following steps:

Note Although this upgrade method works, it is less efficient than other methods of upgrading the router's software.

Procedure

Step 1 Copy each individual subpackage and the provisioning file into the bootflash: directory using the **copy** command. Note that this method of running the router will only work if all the individual subpackages for a release and a provisioning file are downloaded onto the router and stored in the bootflash: directory. No other file directories should be used for booting the router using individual subpackages.

The files can also be moved on the router physically using a USB Flash drive.

Step 2 Configure the router to boot using the provisioning file.

The sequence below provides an example that describes how to boot the router using the provisioning file named packages.conf that was stored with the other subpackages in the bootflash: file system. The router runs using individual subpackages once the reload is complete.

```
Router(config)# no boot system
Router(config)# config-register 0x2102
Router(config)# boot system flash bootflash:packages.conf
Router(config)# exit
*May 11 01:31:04.815: %SYS-5-CONFIG_I: Configured from console by con
Router# write memory Building configuration... [OK]
Router# reload
```

Installing an Optional SubPackage

To run the router using an optional subpackage, perform the following steps for each Supervisor in the chassis:

Procedure

Step 1 Verify that the Supervisor is running in individual subpackage mode and was booted from a provisioning file.

Step 2 Verify that the version of the optional subpackage that you want to install is the same version as the software running on the active Supervisor.

Step 3 Download the optional subpackage that you want to install. Optional subpackages must be downloaded independently from consolidated packages for the Cisco cBR Series Routers.

Step 4 On each Supervisor, copy the optional subpackage to the directory where any other individual subpackages and the provisioning file is located.

Step 5 Run the **request platform software package install rp file** command, as shown in the following example.

Note Do not use the optional slot or bay keywords for the initial installation.

```
Router# request platform software package install rp 0 file
bootflash: cbrsup-universal*.bin
--- Starting local lock acquisition on R0 --- Finished local lock acquisition on R0

--- Starting file path checking --- Finished file path checking

--- Starting image file verification --- Checking image file names Verifying image file
locations Locating image files and validating name syntax
Found cbrsup-universal*.bin
Inspecting image file types Processing image file constraints Creating candidate provisioning
 file

WARNING: No package of type sipspawmak9 is installed.
WARNING: Package will be installed for all SIP slots and bays. Finished image file
verification
--- Starting candidate package set construction --- Verifying existing software set Processing
 candidate provisioning file Constructing working set for candidate package set Constructing
 working set for running package set Checking command output Constructing merge of running
 and candidate packages Checking if resulting candidate package set would be complete
Finished candidate package set construction

--- Starting compatibility testing ---
Determining whether candidate package set is compatible

WARNING:
WARNING: Candidate software combination not found in compatibility database
WARNING:

Determining whether installation is valid

WARNING:
WARNING: Candidate software combination not found in compatibility database
WARNING:

Software sets are identified as compatible Checking IPC compatibility with running software
 Checking candidate package set infrastructure compatibility Checking infrastructure
compatibility with running software Checking package specific compatibility Finished
compatibility testing

--- Starting impact testing --- Checking operational impact of change Finished impact testing

--- Starting list of software package changes --- No old package files removed New files
list:
Added cbrsup-universal*.bin Finished list of software package changes

--- Starting commit of software changes --- Updating provisioning rollback files Creating
pending provisioning file Committing provisioning file Finished commit of software changes

--- Starting analysis of software changes --- Finished analysis of software changes

--- Starting update running software --- Blocking peer synchronization of operating
information Creating the command set placeholder directory
Finding latest command set
Finding latest command shortlist lookup file Finding latest command shortlist file Assembling
 CLI output libraries
Assembling CLI input libraries
Applying interim IPC and database definitions
Replacing running software Replacing CLI software Restarting software
Restarting software: target frus filtered out ... skipped
Applying final IPC and database definitions Generating software version information Notifying
 running software of updates
```

```
Unblocking peer synchronization of operating information Unmounting old packages
Cleaning temporary installation files
Finished update running software

SUCCESS: Finished installing software.
```

Upgrading Individual SubPackages

Patch Installation

Individual subpackages can be upgraded in subpackage mode. Patch releases consist of one or more subpackages. After a patch has been installed, a message is displayed indicating whether the entire chassis or only the line cards must be rebooted for the updates to take effect.

Installing a Patch that Affects Both Line Card and Supervisor Card

Procedure

Step 1 The Cisco cBR router must be in subpackage mode.

Step 2 Copy the patch file to the same location as the active subpackages on both the active and standby supervisor cards.

Step 3 Install the patch using the **request platform software package install rp** *slot* **file** *patch file* command on the standby card.

Step 4 Install the patch using the**request platform software package install rp** *slot* **file** *patch file* command on the active card.

Step 5 Reload the chassis.

Installing a Patch that Affects Only Line Cards

Procedure

Step 1 The Cisco cBR router must be in subpackage mode.

Step 2 Copy the patch file to the same location as the active subpackages on both the active and standby supervisor cards.

Step 3 Install the patch using the **request platform software package install rp** *slot* **file** *patch file* command on the standby card.

Step 4 Install the patch using the**request platform software package install rp** *slot* **file** *patch file* command on the active card.

Step 5 Reload the line cards.

Installing a Patch that Affects Only Supervisor Cards

Procedure

Step 1 The Cisco cBR router must be in subpackage mode.

Step 2 Copy the patch file to the same location as the active subpackages on both the active and standby supervisor cards.

Step 3 Install the patch using the **request platform software package install rp** *slot* **file** *patch file* command on the standby card.

Step 4 Switch over to the standby card. This will make it the active card.

Step 5 Install the patch using the **request platform software package install rp** *slot* **file** *patch file* command on the standby card.

Upgrading a Line Card SubPackage

Use the **request platform software package install node file** *filename* command to upgrade a line card subpackage.

```
Router# request platform software package install node file
bootflash:/subpkg/cbr_patch.5.0.tar

NOTE: Currently node has booted from a provisioning file
NOTE: Going to start a dual rp sub-packages node ISSU install

--- Starting initial file path checking ---
Copying bootflash:/subpkg/cbr_patch.5.0.tar to stby-bootflash:/subpkg/cbr_patch.5.0.tar
Finished initial file path checking

--- Starting config-register verfication ---
Finished config-register verfication

--- Starting image file expansion ---
Expanding image file: bootflash:/subpkg/cbr_patch.5.0.tar
Image file expanded and copied
Expanding image file: stby-bootflash:/subpkg/cbr_patch.5.0.tar
Image file expanded and copied
Finished image file expansion


STAGE 1: Installing software on standby RP
==========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0

--- Starting installation state synchronization ---
Finished installation state synchronization

--- Starting local lock acquisition on R1 ---
```

```
Finished local lock acquisition on R1

--- Starting file path checking ---
Finished file path checking

--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
  Found cbrsup-clcdocsis.2015-02-20_01.02.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification

--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction

--- Starting ISSU compatiblity verficiation ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatiblity verficiation

--- Starting list of software package changes ---
Old files list:
  Removed cbrsup-clcdocsis.BLD_V155_2_S_XE315_THROTTLE_LATEST_20150217_110041-st
d.SSA.pkg
New files list:
  Added cbrsup-clcdocsis.2015-02-20_01.02.SSA.pkg
Finished list of software package changes

--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
Committing provisioning file
Finished commit of software changes

--- Starting analysis of software changes ---
Finished analysis of software changes

--- Starting update running software ---
Blocking peer synchronization of operating information
Creating the command set placeholder directory
  Finding latest command set
  Finding latest command shortlist lookup file
  Finding latest command shortlist file
  Assembling CLI output libraries
  Assembling CLI input libraries
  Assembling Dynamic configuration files
  Applying interim IPC and database definitions
  Replacing running software
  Replacing CLI software
  Restarting software
  Restarting software: target frus filtered out ... skipped
```

```
  Applying final IPC and database definitions
  Generating software version information
  Notifying running software of updates
  Unblocking peer synchronization of operating information
Unmounting old packages
Cleaning temporary installation files
  Finished update running software

SUCCESS: Finished installing software.

STAGE 2: Installing software on active RP
=========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0

--- Starting installation state synchronization ---
Finished installation state synchronization

--- Starting file path checking ---
Finished file path checking

--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
  Found cbrsup-clcdocsis.2015-02-20_01.02.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification

--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction

--- Starting ISSU compatiblity verficiation ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatiblity verficiation

--- Starting impact testing ---
Checking operational impact of change
Finished impact testing

--- Starting list of software package changes ---
Old files list:
  Removed cbrsup-clcdocsis.BLD_V155_2_S_XE315_THROTTLE_LATEST_20150217_110041-st
d.SSA.pkg
New files list:
  Added cbrsup-clcdocsis.2015-02-20_01.02.SSA.pkg
Finished list of software package changes

--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
```

```
Committing provisioning file
Finished commit of software changes

--- Starting analysis of software changes ---
Finished analysis of software changes

--- Starting update running software ---
Blocking peer synchronization of operating information
Creating the command set placeholder directory
  Finding latest command set
  Finding latest command shortlist lookup file
  Finding latest command shortlist file
  Assembling CLI output libraries
  Assembling CLI input libraries
  Assembling Dynamic configuration files
  Applying interim IPC and database definitions
  Replacing running software
  Replacing CLI software
  Restarting software
  Restarting software: target frus filtered out ... skipped
  Applying final IPC and database definitions
  Generating software version information
  Notifying running software of updates
  Unblocking peer synchronization of operating information
Unmounting old packages
Cleaning temporary installation files
  Finished update running software

SUCCESS: Finished installing software.
Found clc package
Found clc package
Found clcdocsis package
SUCCESS: Reload Cable Linecard at slot 1
SUCCESS: Reload Cable Linecard at slot 2
SUCCESS: node ISSU finished successfully.
Invoking cleanup routine
```

Use the **request platform software package install node file** *filename* **noreload linecard** command to upgrade a line card subpackage.

```
Router#request platform software package install node file bootflash:/subpkg/cbr_patch.5.0.tar
 noreload linecard
NOTE: Currently node has booted from a provisioning file
NOTE: Going to start a dual rp sub-packages node ISSU install

--- Starting initial file path checking ---
Copying bootflash:/subpkg/cbr_patch.5.0.tar to stby-bootflash:/subpkg/cbr_patch.
5.0.tar
Finished initial file path checking

--- Starting config-register verfication ---
Finished config-register verfication

--- Starting Checking noreload options ---
Finished Checking noreload options

--- Starting image file expansion ---
Expanding image file: bootflash:/subpkg/cbr_patch.5.0.tar
Image file expanded and copied
Expanding image file: stby-bootflash:/subpkg/cbr_patch.5.0.tar
Image file expanded and copied
```

```
Finished image file expansion

Found clc package

STAGE 1: Installing software on standby RP
==========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0

--- Starting installation state synchronization ---
Finished installation state synchronization

--- Starting local lock acquisition on R1 ---
Finished local lock acquisition on R1

--- Starting file path checking ---
Finished file path checking

--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
  Found cbrsup-clcdocsis.2015-03-01_01.40.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification

--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction

--- Starting ISSU compatiblity verficiation ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatiblity verficiation

--- Starting list of software package changes ---
Old files list:
  Removed cbrsup-clcdocsis.2015-03-01_03.43.SSA.pkg
New files list:
  Added cbrsup-clcdocsis.2015-03-01_01.40.SSA.pkg
Finished list of software package changes

--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
Committing provisioning file
Finished commit of software changes

--- Starting analysis of software changes ---
Finished analysis of software changes

--- Starting update running software ---
Blocking peer synchronization of operating information
```

```
Creating the command set placeholder directory
  Finding latest command set
  Finding latest command shortlist lookup file
  Finding latest command shortlist file
  Assembling CLI output libraries
  Assembling CLI input libraries
  Assembling Dynamic configuration files
  Applying interim IPC and database definitions
  Replacing running software
  Replacing CLI software
  Restarting software
  Restarting software: target frus filtered out ... skipped
  Applying final IPC and database definitions
  Generating software version information
  Notifying running software of updates
  Unblocking peer synchronization of operating information
Unmounting old packages
Cleaning temporary installation files
  Finished update running software

SUCCESS: Finished installing software.

STAGE 2: Installing software on active RP
========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0

--- Starting installation state synchronization ---
Finished installation state synchronization

--- Starting file path checking ---
Finished file path checking

--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
  Found cbrsup-clcdocsis.2015-03-01_01.40.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification

--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction

--- Starting ISSU compatiblity verficiation ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatiblity verficiation

--- Starting impact testing ---
Checking operational impact of change
Finished impact testing
```

```
--- Starting list of software package changes ---
Old files list:
  Removed cbrsup-clcdocsis.2015-03-01_03.43.SSA.pkg
New files list:
  Added cbrsup-clcdocsis.2015-03-01_01.40.SSA.pkg
Finished list of software package changes

--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
Committing provisioning file
Finished commit of software changes

--- Starting analysis of software changes ---
Finished analysis of software changes

--- Starting update running software ---
Blocking peer synchronization of operating information
Creating the command set placeholder directory
  Finding latest command set
  Finding latest command shortlist lookup file
  Finding latest command shortlist file
  Assembling CLI output libraries
  Assembling CLI input libraries
  Assembling Dynamic configuration files
  Applying interim IPC and database definitions


  Replacing running software
  Replacing CLI software
  Restarting software
  Restarting software: target frus filtered out ... skipped
  Applying final IPC and database definitions
  Generating software version information
  Notifying running software of updates
  Unblocking peer synchronization of operating information
Unmounting old packages
Cleaning temporary installation files
  Finished update running software

SUCCESS: Finished installing software.
Found clc package
SUCCESS: node ISSU finished successfully.
Invoking cleanup routine
```

Use the **show platform software patch** *n* **info** command to verify completion of this upgrade.

```
Router#show platform software patch 1 info
cbrsup-clciosdb: 3.15 (0.0)
cbrsup-clc-firmware: 3.15 (0.0)
cbrsup-clcvideo: 3.15 (0.0)
cbrsup-clcios: 3.15 (0.0)
cbrsup-clccontrol: 3.15 (0.0)
cbrsup-clcdocsis: 3.15 (1.0)
cbrsup-clcmipsbase: 3.15 (0.0)


Router#show platform software patch 2 info
cbrsup-clciosdb: 3.15 (0.0)
cbrsup-clc-firmware: 3.15 (0.0)
```

```
cbrsup-clcvideo: 3.15 (0.0)
cbrsup-clcios: 3.15 (0.0)
cbrsup-clccontrol: 3.15 (0.0)
cbrsup-clcdocsis: 3.15 (1.0)
cbrsup-clcmipsbase: 3.15 (0.0)
```

Use the **show platform software ios** *slot-number* **restart info** command to verify completion of this upgrade. This example shows the output of this **show** command for the RF line card slot number 2.

```
Router#show platform software ios 2 restart info
IOSD process restart info:
  Process restartable: Yes
  IOSD restart state : NOT_RESTARTED_YET
  Total Modem Count  : 251
  Active Modem Count : 251


Router#
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Consolidated Packages and SubPackages Management

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 2: Feature Information for Consolidated Packages and SubPackages Management

Feature Name	Releases	Feature Information
Consolidated Packages and SubPackages Management	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.