

# Action Commands


- clear configuration inconsistency, on page 2
- hw-module location, on page 5
- install add, on page 8
- install activate (IOS XR 64 bit), on page 14
- install deactivate (IOS XR 64 bit), on page 17
- install commit, on page 19
- install upgrade source , on page 21
- install remove, on page 25
- install replace, on page 28
- reload, on page 30
- ztp enable, on page 32
- ztp disable, on page 33
- ztp initiate, on page 34

# clear configuration inconsistency

To clear an inconsistency alarm for a router configuration or admin plane configuration, use the **clear configuration inconsistency** command in Admin EXEC mode or EXEC mode.

**clear configuration inconsistency**

## Syntax Description

This command has no keywords or arguments.

## Command Default

Administration EXEC mode: Clears the inconsistency alarms for the admin plane configuration.

EXEC mode: Clears the inconsistency alarms for an SDR configuration.

## Command Modes

Admin EXEC mode

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

An inconsistency alarm is set when there is a failure to restore the configuration; this can occur during router startup, or when a line card or route switch processor (RSP) card is inserted or removed.

If an inconsistency alarm is set, a message similar to the following example is displayed:

```
RP/0/0/CPU0:May 26 11:58:40.662 : cfgmgr-rp[130]: %MGBL-CONFIGCLI-3
   BATCH_CONFIG_FAIL : 28 config(s) failed during startup. To view
   failed config(s) use the command - "show configuration failed startup"

RP/0/0/CPU0:May 26 11:58:41.731 : cfgmgr-rp[130]:
   %MGBL-CONFIG-3-ADMIN_INCONSISTENCY_ALARM : Admin plane configuration
   inconsistency alarm has been raised. Configuration commits will be
   blocked until an ADMIN plane 'clear configuration inconsistency' command
   has been run to synchronize persisted admin plane configuration with
   running admin configuration.
```

When the inconsistency alarm is set, all configuration commit operations fail until the alarm is cleared using the **clear configuration inconsistency** command. This command clears the alarm and removes the failed configuration.

For example, the following configuration commit fails to finish due to an existing inconsistency alarm:

```
RP/0/RSP0/CPU0:router# configure

ADMIN plane running configuration is inconsistent with persistent
  configuration.
No configuration commits will be allowed until an admin plane
'clear configuration inconsistency' command is performed.
RP/0/RSP0/CPU0:router(config)# hostname router2
RP/0/RSP0/CPU0:router(config)#commit
```

```
ADMIN plane running configuration is inconsistent with persistent
  configuration.
No configuration commits will be allowed until an admin plane
'clear configuration inconsistency' command is performed.
```

Enter the **clear configuration inconsistency** command to clear the alarm and allow commit operations to continue.

**Note** To reapply the failed configuration, you must reapply and recommit the configuration. Use the **load configuration failed** command with the **startup** keyword to populate the target configuration with the contents of the previous failed configuration from the startup configuration.

Use the **show configuration history** command with the **alarm** keyword to view the inconsistency alarm set and alarm clear events in the configuration history log.

## Command Modes

To clear the inconsistency alarms for the admin plane configuration, enter the **clear configuration inconsistency** command in administration EXEC mode.

To clear the inconsistency alarms for the router, enter the **clear configuration inconsistency** command in EXEC mode.

## Task ID

| Task ID | Operations |
|---|---|
| config-services | execute |

The following example shows how to clear the inconsistency alarms for the admin plane configuration by entering the **clear configuration inconsistency** command in administration EXEC mode:

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# clear configuration inconsistency

Creating any missing directories in Configuration File system...OK
Initializing Configuration Version Manager...OK
Syncing ADMIN commit database with running configuration...OK
Re-initializing cache files...OK
Updating Commit Database.  Please wait...[OK]
```

The following example shows how to clear the inconsistency alarms for a router configuration. The command is entered in EXEC mode.

```
RP/0/RSP0/CPU0:router# clear configuration inconsistency

Creating any missing directories in Configuration File system...OK
Initializing Configuration Version Manager...OK
Syncing commit database with running configuration...OK
Re-initializing cache files...OK
Updating Commit Database.  Please wait...[OK]
```

In the following example, a history of the inconsistency alarms set and cleared for the router configuration are displayed using the **show configuration history** command with the **alarm** keyword:

```
RP/0/RSP0/CPU0:router# show configuration history alarm

  Sno.  Event      Info                          Time Stamp
  ~~~~  ~~~~~      ~~~~                          ~~~~~~~~~~
  1     alarm      inconsistency alarm raised    Thu Jun 22 15:23:15 2009
  2     alarm      inconsistency alarm cleared   Thu Jun 22 15:42:30 2009
  3     alarm      inconsistency alarm raised    Sun Jul  9 13:39:57 2009
  4     alarm      inconsistency alarm cleared   Sun Jul  9 14:15:48 2009
  5     alarm      inconsistency alarm raised    Sat Jul 15 18:18:26 2009
  6     alarm      inconsistency alarm cleared   Sat Jul 15 19:21:03 2009
```

# hw-module location

To configure various hardware attributes for a specific node, or for all nodes installed in the router, use the **hw-module location** command in System Admin EXEC mode.

To recover the RP (route processor) and SC (shelf controller) card or all the nodes in a system, use the **hw-module location** command in Sysadmin EXEC mode.

To reset or shutdown a specific node, or to put a node into maintenance mode, use the **hw-module location** command in the  mode.

**hw-module location** *location|all]* **bootmedia** *[recovery-partition|usb|network]* **reloadofflineonlinereload [force]shutdown [force]**

## Syntax Description

| | |
|---|---|
| *node-id|all* | Node whose hardware attributes you want to configure. The *node-id* is expressed in the *rack/slot/module* notation in the mode.<br>To confgure all nodes, use *all* .<br>**Note** Enter the **show platform** command to see the location of all nodes installed in the router. |
| **bootmedia** | Select a boot media to load images. |
| **offline** | Changes the state of the hardware module to offline to perform diagnostics. When card is set to offline, it is taken out of the network. However, the card is powered on and maintains connectivity so that it can be accessed for diagnostics. |
| **online** | Changes the state of the hardware module to online for normal operation of the device. |
| **reload** | Reloads the hardware module. |
| **shutdown** | shut down the hardware module. |

## Command Default

None

## Command Modes

System Admin EXEC

## Command History

| Release | Modification |
|---|---|
| Release 6.0 | This command was introduced. |

## Usage Guidelines

The **hw-module location***all***bootmedia** is used to reimage a node or all nodes.

When a router reloads using **hw-module location** *all*, if the line card experiences a timeout waiting for the sysdb configuration replay, the system will notify the user once to reload the line card. This reload allows the sysdb configuration to come up properly after the line card restarts.

If the **bootmedia network** command specifies a single location, the node will get reloaded and reimaged with an image downloaded from the master RP.

**Note** The **bootmedia network** option is applicable to RP and LC nodes. It is not applicable to FC and SC nodes.

The master RP can be determined by looking at the **show controller card-mgr inventory summary** command output in mode:

```
sysadmin-vm:0_RP0#  show controller card-mgr inventory summary
Fri Feb  26 03:24:22.205 UTC-08:00
Card Manager Inventory Summary :
                                BP                  HW
Location Card Type              ID  Serial Number   Ver    Card State
-------------------------------------------------------------------------------
0/0      NC55-24x100G-SE        1   SAL1934MNEX     0.103  CARD_READY
0/3      NC55-6X100GE-PROT      4   SAL1915D4D9     0.110  CARD_READY
0/FC1    NC55-5508-FC           22  SAL1923GDG2     0.305  CARD_READY
0/FC3    NC55-5508-FC           24  SAL1926HZYB     0.106  CARD_READY
0/FC5    NC55-5508-FC           26  SAL1911B7WJ     0.303  CARD_READY
0/RP0    NC55-RP (Master)       27  SAL1925HFTH     1.1    CARD_READY
0/RP1    NC55-RP (Slave)        28  SAL1924GUZC     1.1    CARD_READY
0/SC0    NC55-SC (Slave)        29  SAL190389WF     1.4    CARD_READY
0/SC1    NC55-SC (Master)       30  SAL1923G36G     1.4    CARD_READY
```

If **bootmedia network** specifies the master RP location, the master RP will reboot and get reimaged with an image downloaded from an external PXE server via the master RP's management ethernet interface.

If **bootmedia network** command specifies *all* locations, the master RP will reboot and get reimaged with an image downloaded from an external PXE server via the master RP's management ethernet interface. After the master RP is reimaged, all other cards will get reloaded and reimaged with an image downloaded from the master RP.

The **bootmedia usb** option is only available for RP0 and RP1. When this option is selected, the RP will reload and be reimaged using the image stored on the RP's usb device.

```
sysadmin-vm:0_RP0# hw-module location 0/RP1 ?
Possible completions:
  bootmedia   Select boot media to load image from
  offline     Take a hardware module offline for diagnostics
  online      Take a hardware module online for normal operation
  reload      Reload a hardware module
  shutdown    Shut down a hardware module
```

```
sysadmin-vm:0_RP0# hw-module location 0/3 bootmedia ?
Possible completions:
  network

sysadmin-vm:0_RP0# hw-module location 0/RP1 bootmedia ?
Possible completions:
  network  usb

sysadmin-vm:0_RP0# hw-module location 0/RP1 bootmedia usb ?
Possible completions:
  reload   Reload a hardware module

sysadmin-vm:0_RP0# hw-module location 0/RP1 bootmedia usb reload ?
Possible completions:
  |      Output modifiers
  <cr>

sysadmin-vm:0_RP0# hw-module location 0/3 bootmedia network reload
Fri Feb  26 00:47:26.972 UTC-08:00
Reload hardware module ? [no,yes] yes
result Card reload request on 0/3 succeeded.
sysadmin-vm:0_RP0#
```

# install add

To copy the contents of a package installation envelope (PIE) file to a storage device, use the **install add** command in EXEC or  Admin EXEC mode EXEC mode.

Administration EXEC Mode:

**install add** [**source** *source-path* | **tar**] *file* [**activate** [**pause sw-change**] [**auto-abort-timer** *time*] [**location** *node-id*]] [**asynchronous** | **synchronous**] [**parallel-reload**] [**prompt-level** {**default** | **none**}] [**if-active**] [**sdr** *sdr-name*]

EXEC Mode:

**install add** [**source** *source-path* | **tar**] *file* [**activate** [**pause sw-change**] [**auto-abort-timer** *time*] [**location** *node-id*]] [**asynchronous** | **synchronous**] [**parallel-reload**] [**prompt-level** {**default** | **none**}]

## Syntax Description

| | |
|---|---|
| **source** *source-path* | (Optional) Specifies the source location of the PIE files to be appended to the PIE filenames. Location options are as follows:<br>• **disk0:**<br>• **disk1:**<br>• **compactflash:**<br>• **harddisk:**<br>• **ftp:**//*username* **:***password*@*hostname* or *ip-address*/*directory-path*<br>• **rcp:**//*username*@*hostname* or *ip-address*/*directory-path*<br>• **tftp:**//*hostname* or *ip-address*/*directory-path* |
| **tar** | (Optional) Indicates that the PIE file is contained in a tar file. |
| *file* | Name and location of the PIE file (composite package) to install. If a source path location is specified using the **source** keyword, the *file* argument can be either a fully specified PIE file path, or a path to the PIE file relative to the source path.<br>**Note**<br>Up to 32 PIE files can be added to a device in a single **install add** operation.<br>If the **tar** keyword is used, the *file* argument is a tar file that contains one or more PIE files, or directories containing PIE files. Up to 16 tar files can be added, out of the possible 32 install files. |
| **activate** | (Optional) Activates the package or packages. This option is run only if the **install add** operation is successful. |

| | |
|---|---|
| **pause sw-change** | (Optional) Pauses the operation before locking the configuration for the software activation. While the operation is paused, you can perform configuration changes. You control the resumption of the operation at the CLI prompt. |
| **auto-abort-timer** *time* | (Optional) Specifies an abort timer value, *time*, in minutes, which when expired loads the last committed loadpath. |
| **location** *node-id* | (Optional) Activates a package on the designated node. The *node-id* argument is expressed in *rack/slot/module* notation.<br>**Note**<br>A package cannot be activated on a single node unless some version of the package being activated is already active on all nodes. For example, a Multiprotocol Label Switching (MPLS) package cannot be active on only one node. If a version of the MPLS package is already active on all nodes, an MPLS package then could be upgraded or downgraded on a single node. |
| **asynchronous** | (Optional) Performs the command in asynchronous mode. In asynchronous mode, this command runs in the background, and the EXEC prompt is returned as soon as possible. This is the default mode. |
| **synchronous** | (Optional) Performs the command in synchronous mode. This mode allows the installation process to finish before the prompt is returned. |
| **parallel-reload** | (Optional) Forces all cards on the router to reload at the same time and then come up with the new software, rather than proceeding according to the option encoded in the install package. |
| **prompt-level** {**default** \| **none**} | (Optional) Specifies when you are prompted for input during the procedure.<br>• **default**—You are prompted only when input is required by the operation.<br>• **none**—You are never prompted. |
| **if-active** | (Optional. Administration EXEC mode only.) Activates the optional packages only if a version is already active. |
| **sdr** *sdr-name* | (Optional. Administration EXEC mode only.) Activates a package for a specific secure domain router (SDR). The *sdr-name* argument is the name assigned to the SDR. The only SDR available is Owner, which refers to the entire router. |

## Command Default

Packages are added to the storage device, but are not activated.

The operation is performed in asynchronous mode. The **install add** command runs in the background, and the EXEC prompt is returned as soon as possible.

**Command Modes**

EXEC

EXEC mode

Admin EXEC mode

**Command History**

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |
| Release 4.0.0 | This command was removed from EXEC mode.<br>The **sdr** keyword was removed. |

**Usage Guidelines**

Use the **install add** command to unpack the package software files from a PIE file and copy them to the boot device (usually disk0:) .

- From administration EXEC mode, the package software files are added to all route processors (RPs) installed in the router. If the **install add** command is entered without specifying an SDR, then the package files are added to all RPs in the owner SDR.
- From EXEC mode, the package software files are added to the RPs only for the SDR to which you are logged in.

### Adding and Activating a Package

Software packages remain inactive until activated with the **install activate** command.

To add and activate a package at the same time, use the **install add** command with the **activate** keyword. When this command is used, the keywords and rules for package activation apply.

- To add and activate a package for the owner SDR, enter the **install add** command with the **activate** keyword from administration EXEC mode.
- It is also possible to add and activate a package using the **install add** command with the **activate** keyword from EXEC mode.

**Note** SDR-specific activation is supported for specific packages and upgrades, such as optional packages and Software Maintenance Upgrades (SMUs). Packages that do not support SDR-specific activation can be activated for all SDRs simultaneously only from administration EXEC mode. For detailed instructions, see the *Managing Cisco IOS XR Software Packages* module of *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.

**Note** If a software activation requires a node reload, the config-register for that node should be set to autoboot. If the config-register for the node is not set to autoboot, then the system automatically changes the setting and the node reloads. A message describing the change is displayed.

### Synchronous Mode

Use the **install add** command with the **synchronous** keyword to complete the operation before the prompt is returned. A progress bar indicates the status of the operation. For example:

```
- 1% complete: The operation can still be aborted (ctrl-c for options)
\ 10% complete: The operation can still be aborted (ctrl-c for options)
```

### TFTP Services and Image Size

Some Cisco IOS XR images may be larger than 32 MB, and the TFTP services provided by some vendors may not support a file this large. If you do not have access to a TFTP server that supports files larger than 32 MB:

- Download the software image using FTP or rcp.
- Use a third-party or freeware TFTP server that supports file sizes larger than 32 MB.

### Adding tar Files

Use the **tar** keyword to add one or more PIE files in the tar file format. If the **tar** keyword is used, only a single tar file can be added.



**Note** Multiple tar files or a combination of PIE and tar files is not supported.

Note the following regarding tar files:

- The *file* argument must include the complete location of the tar file.
- The tar file can contain only PIE files and directories containing PIE files. For example:
  - The tar file pies.tar containing the files x.tar and y.pie fails because x.tar is not a PIE file.
  - The tar file pies.tar containing the file x.pie and the directory dir_a, where dir_a contains a PIE file y.pie succeeds.
  - The tar file pies.tar containing the file x.pie and the directory dir_a, where dir_a contains a tar file y.tar fails because y.tar is not a PIE file.
  - The tar file pies.tar containing the PIE files x.pie, y.pie, ...*.pie succeeds.
- The **source** keyword is not supported with the **tar** keyword.

Following is a valid example of using the **tar** keyword:

```
RP/0/RSP0/CPU0:router(admin)# install add tar
tftp://223.255.254.254/install/files/pies.tar
```

You can add and activate tar files at the same time. In other words, the **install add** command is supported using the **tar** and the **activate** keywords simultaneously.

### Adding Multiple Packages

To add multiple PIE files, use the **source** keyword to specify the directory path location of the PIE files. Then list all the PIE filenames, as necessary. This alleviates the need to repeat the directory location for each PIE file. Up to 32 files can be added, of which 16 can be tar files.

Following is an example of the **install add** command using the **source** keyword:

```
RP/0/0/CPU0:router(admin)# install add source
tftp://192.168.201.1/images/myimages/comp-asr9k-mini.pie
asr9k-mgbl-p.pie asr9k-mpls-p.pie
asr9k-mcast-p.pie
```

The following example also illustrates a valid use of the **install add** command with the **source** keyword:

```
RP/0/RSP0/CPU0:router(admin)# install add source
tftp://192.168.254.254/images/user/asr9k-mcast-p.pie
pies/asr9k-mpls-p.pie
ftp://1.2.3.4/other_location/asr9k-mgbl-p.pie
```

In the previous example, three PIE files are added from the following locations:

- tftp://192.168.254.254/images/user/asr9k-mcast-p.pie
- tftp://192.168.254.254/images/user/pies/asr9k-mpls-p.pie
- ftp://1.2.3.4/other_location/asr9k-mgbl-p.pie

### Parallel Reload

Installation operations are activated according to the method encoded in the package being activated. Generally, this method has the least impact for routing and forwarding purposes, but it may not be the fastest method from start to finish and can require user interaction by default. To perform the installation procedure as quickly as possible, you can specify the **parallel-reload** keyword. This forces the installation to perform a parallel reload, so that all cards on the router reload simultaneously, and then come up with the new software. This impacts routing and forwarding, but it ensures that the installation is performed without other issues.

### Pausing Activation Before Configuration Lock

If you specify the **activate** keyword, use the **pause sw-change** keywords to pause the software activation operation before locking the configuration. A software activation operation begins with preparatory steps, such as software checks, and then proceeds with the actual activation of the new software. The configuration is locked for the activation. If you specify the **pause sw-change** keywords, the operation pauses before locking the configuration and provides you with the option to hold the operation while you perform configuration changes, and proceed with the activation whenever you choose. This is useful, for example, if your workflow involves configuring a router out of the network during software installation and you want to minimize the time that the router is out of the network. You can specify these keywords for both asynchronous and synchronous operations. In both cases, follow the onscreen instructions to control the pausing and completion of the operation.

The following example shows how to add a PIE file for all SDRs in the system. In the following example, a Multiprotocol Label Switching (MPLS) package is added in synchronous mode. This operation copies the files required for the package to the storage device. This package remains inactive until it is activated with the **install activate** command.

```
RP/0/RSP0/CPU0:router# admin
```

```
RP/0/RSP0/CPU0:router(admin)# install add
tftp://209.165.201.1/asr9k-mpls.pie synchronous

Install operation 4 'install add /tftp://209.165.201.1/asr9k-mpls.pie synchronous'
  started by user
'user_b' at 03:17:05 UTC Mon Nov 14 2005.
Info:     The following package is now available to be activated:
Info:
Info:         disk0:asr9k-mpls-3.3.80
Info:
Install operation 4 completed successfully at 03:18:30 UTC Mon Nov 14 2005.
```

In the following example, a package is added and activated with a single command:

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# install add disk1:asr9k-mgbl-p.pie-3.8.0activate

Install operation 4 'install add /disk1:asr9k-mgbl-p.pie-3.8.0 activate' started
by user 'user_b' at 07:58:56 UTC Wed Mar 01 2006.
The install operation will continue asynchronously.
:router(admin)#Part 1 of 2 (add software): Started
Info:     The following package is now available to be activated:
Info:
Info:         disk0:asr9k-mgbl-3.8.0
Info:
Part 1 of 2 (add software): Completed successfully
Part 2 of 2 (activate software): Started
Info:     The changes made to software configurations will not be persistent across
system reloads. Use the command 'admin install
Info:     commit' to make changes persistent.
Info:     Please verify that the system is consistent following the software change
using the following commands:
Info:         show system verify
Info:         install verify
Part 2 of 2 (activate software): Completed successfully
Part 1 of 2 (add software): Completed successfully
Part 2 of 2 (activate software): Completed successfully
Install operation 4 completed successfully at 08:00:24 UTC Wed Mar 01 2006.
```

# install activate (IOS XR 64 bit)

To enable the package configurations to be made active on the router so new features and software fixes take effect, use the **install activate** command in EXEC mode or Admin EXEC mode.

**install activate** *package_name*
**install activate id** *operation_id*

## Syntax Description

| | |
|---|---|
| *package_name* | Enter the package names separated by space.<br>**Note** Up to 16 packages can be specified in a single **install activate** command at a time. Multiple packages can be specified using the wildcard syntax, for example, asr9k-*-x64-*3I. |
| **id** *operation_id* | The *operation_id* is the ID from the **install add** operation.<br>The **show install request** command displays the operation id number of the **install add** operation and its status. You can also find the *operation_id* in the **show install log** command output. |

## Command Default

The **install activate** command activates all packages that were added in the specified **install add** operation and the operation is performed in an asynchronous mode. The command runs in the background and the EXEC prompt is returned soon after.

If you use the operation ID (from the add operation) to activate packages, all packages that were added in the specified install add operation are activated together. You do not have to activate the packages individually.

For example, if five packages are added in operation 6, all the five packages are activated together by executing **install activate id** *6* command.

**Note**

- Activation takes some time and does not happen instantaneously.
- Activation of some SMUs require a manual reloading of the router. When such SMUs are activated, a warning message is displayed to perform reload. The components of the SMU get activated only after the reload is complete.

## Command Modes

EXEC mode

Admin EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 6.1.2 | Support for IOS XR 64 bit **install activate** command was added. |

## Usage Guidelines

- Only inactive packages can be activated. Use the **show install inactive** command to identify the inactive packages that are present in the repository.
- If you want to activate packages using the **install activate id** *operation_id* command syntax, use the **show install log** command to identify the operation ID of the add operation.

When a router reloads using **install activate**, if the line card experiences a timeout waiting for the sysdb configuration replay, the system will notify the user once to reload the line card. This reload allows the sysdb configuration to come up properly after the line card restarts.

The following example lets you activate packages by specifying the package names:

```
RP/0/RSP0/CPU0:router# install activate asr9k-m2m-x64-2.0.0.0-r61106I.x86_64
asr9k-optic-x64-1.0.0.0-r61106I.x86_64

Jun 22 14:09:25 Package list:
Jun 22 14:09:25 asr9k-m2m-x64-2.0.0.0-r61106I.x86_64
Jun 22 14:09:25 asr9k-optic-x64-1.0.0.0-r61106I.x86_64
Jun 22 14:09:35 Install operation will continue in the background

RP/0/RSP0/CPU0:router#
```

The following example lets you activate packages by specifying the id from the add operation:

```
RP/0/RSP0/CPU0:router# install activate id 6

Jun 22 15:02:24 Package list:
Jun 22 15:02:24 asr9k-bgp-x64-1.0.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-isis-x64-1.0.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-k9sec-x64-1.1.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-li-x64-1.1.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-parser-x64-1.0.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-m2m-x64-2.0.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-mgbl-x64-2.0.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-optic-x64-1.0.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-mcast-x64-1.1.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-mpls-te-rsvp-x64-1.0.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-eigrp-x64-1.0.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-ospf-x64-1.0.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-mpls-x64-2.0.0.0-r61106I.x86_64
Jun 22 15:02:25 Skipped packages which were already active:
Jun 22 15:02:25 asr9k-bgp-x64-1.0.0.0-r61106I.x86_64
Jun 22 15:02:25 asr9k-m2m-x64-2.0.0.0-r61106I.x86_64
Jun 22 15:02:25 asr9k-optic-x64-1.0.0.0-r61106I.x86_64
Jun 22 15:02:33 Install operation will continue in the background

RP/0/RSP0/CPU0:router#
```

The following example lets you activate multiple packages using the wildcard syntax:

```
RP/0/RSP0/CPU0:router#install activate asr9k-*-x64-*3I

Jun 16 19:35:06 Install operation 105 started by root:
install activate pkg asr9k-*-x64-*3I
Jun 16 19:35:06 Package list:
```

```
Jun 16 19:35:06 asr9k-eigrp-x64-1.0.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-ospf-x64-1.0.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-m2m-x64-2.0.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-k9sec-x64-1.1.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-mpls-x64-1.1.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-bgp-x64-1.0.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-isis-x64-1.0.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-mini-x64-6.1.1.03I
Jun 16 19:35:06 asr9k-mgbl-x64-2.0.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-parser-x64-1.0.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-optic-x64-1.0.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-mcast-x64-1.1.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-mpls-te-rsvp-x64-1.0.0.0-r61103I.x86_64
Jun 16 19:35:06 asr9k-li-x64-1.0.0.0-r61103I.x86_64
Jun 22 15:02:33 Install operation will continue in the background

RP/0/RSP0/CPU0:router#
```

# install deactivate (IOS XR 64 bit)

To remove a package from the active software set, use the **install deactivate** command in EXEC mode or Admin EXEC mode.

**install deactivate** *package_name*
**install deactivate id** *operation_id*

## Syntax Description

| | |
|---|---|
| *package_name* | Enter the package names separated by space.<br>**Note** Up to 16 packages can be specified in a single **install deactivate** command at a time. Multiple packages can be specified using the wildcard syntax, for example, asr9k-*-x64-*3I. |
| **id** *operation_id* | The *operation_id* is the ID from the **install add** operation.<br>The **show install request** command displays the operation id number of the **install add** operation and its status. You can also find the *operation_id* in the **show install log** command output. |

## Command Default

The **install deactivate** command deactivates all features and software patches associated with the specified activated packages. This operation is performed in an asynchronous mode and the command runs in the background.

If you use the operation ID (from the add operation) to deactivate packages, all packages that were added in the specified **install add** operation are deactivated together. You do not have to deactivate the packages individually.

For example, if five packages are added in operation 6, all the five packages are deactivated together by executing **install deactivate id** *6* command.

**Note** The System admin packages that were added as a part of the **install add** operation (of the ID used in deactivate operation) will also be deactivated.

## Command Modes

EXEC mode

Admin EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 6.1.2 | Support for IOS XR 64 bit **install deactivate** command was added. |

## Usage Guidelines

- Only active packages can be deactivated. Use the **show install active** command to identify the active packages.

- If you want to deactivate packages using the **install deactivate id** *operation_id* command syntax, use the **show install log** command to identify the operation ID of the add operation.
- If you want to remove the inactive packages from the repository, use the **show install inactive** command to identify the deactivated packages that are now listed as inactive packages.

  Then, use the **install remove** command to remove the packages from the repository.

  When a router reloads using **install deactivate**, if the line card experiences a timeout waiting for the sysdb configuration replay, the system will notify the user once to reload the line card. This reload allows the sysdb configuration to come up properly after the line card restarts.

The following example lets you deactivate packages by specifying the package names:

```
RP/0/RSP0/CPU0:router# install deactivate asr9k-m2m-x64-2.0.0.0-r61106I.x86_64
asr9k-optic-x64-1.0.0.0-r61106I.x86_64

Jun 22 14:09:25 Package list:
Jun 22 14:09:25 asr9k-m2m-x64-2.0.0.0-r61106I.x86_64
Jun 22 14:09:25 asr9k-optic-x64-1.0.0.0-r61106I.x86_64
Jun 22 14:09:35 Install operation will continue in the background

RP/0/RSP0/CPU0:router#
```

The following example lets you deactivate packages by specifying the id from the add operation:

```
RP/0/RSP0/CPU0:router# install deactivate id 6

Jun 22 15:02:24 Package list:
Jun 22 15:02:24 asr9k-bgp-x64-1.0.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-isis-x64-1.0.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-k9sec-x64-1.1.0.0-r61106I.x86_64
Jun 22 15:02:24 asr9k-li-x64-1.1.0.0-r61106I.x86_64
Jun 22 15:02:33 Install operation will continue in the background

RP/0/RSP0/CPU0:router#
```

The following example lets you deactivate multiple packages using the wildcard syntax:

```
RP/0/RSP0/CPU0:router# install deactivate *-r61103I

Jun 16 19:35:06 Install operation 108 started by root:
install deactivate pkg asr9k-mpls-x64-1.1.0.0-r61103I asr9k-mpls-te-rsvp-x64-1.0.0.0-r61103I

asr9k-mcast-x64-1.1.0.0-r61103I
Jun 16 19:35:06 Package list:
Jun 16 19:35:06 asr9k-mpls-x64-1.1.0.0-r61103I
Jun 16 19:35:06 asr9k-mpls-te-rsvp-x64-1.0.0.0-r61103I
Jun 16 19:35:06 asr9k-mcast-x64-1.1.0.0-r61103I
Jun 16 19:35:06 Install operation will continue in the background

RP/0/RSP0/CPU0:router#
```

# install commit

To save the active software set to be persistent across designated system controller (DSC) reloads, use the **install commit** command in EXEC or Admin EXEC mode EXEC mode.

Administration EXEC Mode:
**install commit** [**location** *node-id* | **sdr** *sdr-name*]
EXEC mode Mode
**install commit**

## Syntax Description

| | |
|---|---|
| **location** *node-id* | (Optional. Admin EXEC mode mode only.)Specifies a node. The *node-id* argument is expressed in *rack*/*slot*/*module* notation. |
| **sdr** *sdr-name* | (Optional. Admin EXEC mode only.) Commits the active software set for a specific SDR. The *sdr-name* argument is the name assigned to the SDR. The only SDR available is Owner, which refers to the entire router. |

## Command Default

Admin EXEC mode: Commits the active software set for all SDRs.

EXEC mode: Commits the active software set for the current SDR.

## Command Modes

EXEC

Admin EXEC mode

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |
| Release 4.0.0 | This command was removed from EXEC mode.<br>Support for the **sdr** keyword was removed. |

## Usage Guidelines

When a package is activated, it becomes part of the current running configuration. To make the package activation persistent across designated secure domain router shelf controller (DSDRSC) reloads, enter the **install commit** command. On startup, the DSDRSC of the SDR loads this committed software set.

If the system is restarted before the active software set is saved with the **install commit** command, the previously committed software set is used.

## Command Modes

To commit the active software set for the owner SDR, use the **install commit** command in either administration EXEC or EXEC mode.

## Task ID

| Task ID | Operations |
|---|---|
| pkg-mgmt | read, write |

The following example shows how to make the current active software set persistent across DSDRSC reloads for all SDRs in the system:

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# install commit

Install operation 16 'install commit' started by user 'user_b' at 19:18:58 UTC
Sat Apr 08 2006.
Install operation 16 completed successfully at 19:19:01 UTC Sat Apr 08 2006.
```

# install upgrade source

To upgrade the software package, use the **install upgrade source** command in administration EXEC mode.

**install upgrade source** [**ftp** | **tftp***system-disk*] [**path** *directory-path*] [*file-name* | **version** *version-number* | **packages**] *[source-file]* [**synchronous**] [**pause-reload**]

## Syntax Description

| | |
|---|---|
| **source** | Specify the source location of the PIE files to be appended to the PIE filenames. Location options are as follows:<br>**ftp:** —Copies from an FTP network server. The syntax is **ftp:**[[[//*username* [:*password*]@] *location*]/*directory*]/*filename.*<br>**tftp:** —Copies from a TFTP network server. The syntax is **tftp:**[[//*location*]/*directory*]/*filename*<br>*system disk* —Copies package source from system disk. Location options are as follows:<br>• **harddisk:** —Copies from the hard disk drive file system (if present).<br>• **disk1:** —Copies from disk1: file system. |
| **path** *directory-path* | Specify the storage device and directory for the file search. The search is performed for the specified directory and all subdirectories in that directory tree.<br>The syntax for directory-path is: *device :[/ directory-path]*<br>If a directory path is not specified, then the search is performed in the current directory (a path of . [dot] is assumed). |
| *file-name* | Only for TFTP, with file that contains a list of packages to be installed<br>**Note**<br>Directory listing is not possible |
| **version** *version-number* | Specify the package version that is to be installed |
| **packages** | Specify the package names to install (packages can be *tar* file) |
| *source-file* | Specify the source location of the PIE files on the system |
| **synchronous** | (Optional) Performs the command in synchronous mode. This mode allows the installation process to be completed before the prompt is returned.<br>**Note**<br>By default, installation operations are performed in asynchronous mode. In asynchronous mode, the command will run without expecting any user inputs while holding the prompt. |

| | |
|---|---|
| **pause-reload** | (Optional) Pauses the operation before any reload occurs. The configuration remains locked for the activation. This keyword precedes the following two keywords:<br>• **pause-reload allow-sw-change**—The operation pauses before locking the configuration and provides the option to hold the operation while you perform configuration changes. You can proceed with the activation whenever you choose.<br>• **pause-reload disallow-sw-change**—The operation pauses before reload but this will not allow you to make any configuration changes.<br>**Note**<br>These keywords are applicable for asynchronous and synchronous operations. In both cases, follow onscreen instructions to control the pausing and completion of the operation. |

**Command Default**

By default **install upgrade source** picks active version packages.

**Command Modes**

Administration EXEC

**Command History**

| Release | Modification |
|---|---|
| Release 5.3.2 | This command was introduced. |

**Usage Guidelines**

When a router reloads using **install upgrade source**, if the line card experiences a timeout waiting for the sysdb configuration replay, the system will notify the user once to reload the line card. This reload allows the sysdb configuration to come up properly after the line card restarts.

**FTP**

Use the following options to upgrade the system using FTP as source:

- Only repository without version—It picks delta packages (ignoring optional packages) of the active version; that is, the difference between packages present in the file specified and packages active on the system.
- Repository with version—It picks packages of the specified version. If a mini package of the specified version is found in the file, it will upgrade the system, provided all dependency and package compatibility checks are completed successfully.
- Packages—Specifies the list of packages to add or upgrade the system. This option can be used to add *tar* files.

**TFTP**

Use the following options to upgrade the system using TFTP as source:

- File-name—This option requires the package list to be provided in a file, which can then be used to upgrade the system or update the packages or SMU's. It picks delta packages (ignoring optional packages) of the active version; that is, the difference between packages present in the file specified and packages active on the system.

- File-name with version—It picks packages of the specified version. If a mini package of the specified version is found in the file, it will upgrade the system, provided all dependency and package compatibility checks are completed successfully.
- Packages—Specifies the list of packages to add or upgrade the system. This option can be used to add *tar* files.

**harddisk**

Use the following options to upgrade the system using harddisk as source:

- Only repository without version—It picks delta packages (ignoring optional packages) of the active version; that is, the difference between packages present in the file specified and packages active on the system.
- Repository with version—It picks packages of the specified version. If a mini package of the specified version is found in the file, it will upgrade the system, provided all dependency and package compatibility checks are completed successfully.
- Packages—Specifies the list of packages to add or upgrade the system. This option can be used to add *tar* files.

## Task ID

| Task ID | Operation |
|---|---|
| pkg-mgmt | execute |

## Example

This example shows how to upgrade a package to 5.2.4 version with image asr9k-mini-px.pie-5.2.4 from the FTP repository, using the **install upgrade source** command:

```
RP/0/RSP0/CPU0:router#admin
RP/0/RSP0/CPU0:router(admin)#install upgrade source ftp://10.10.10.10/yum_like_upgrade
asr9k-mini-px.pie-5.2.4 synchronous
```

This example shows how to upgrade a package to 5.3.2 version from the on-system repository, using the **install upgrade source** command:

```
RP/0/RSP0/CPU0:router#admin
RP/0/RSP0/CPU0:router(admin)#install upgrade source harddisk:/images/532  version 5.3.2
synchronous
```

This example shows how to upgrade package to release 5.1.0 from the TFTP repository, using the **install upgrade source** command:

```
RP/0/RSP0/CPU0:router#admin
RP/0/RSP0/CPU0:router(admin)# install upgrade source tftp://10.10.10.10/auto/tftpboot/userid
 file-name packages.txt version 5.1.0  synchronous
```

This example shows how to add and activate the package or SMU of active version using the **install upgrade source** command:

```
RP/0/RSP0/CPU0:router#admin
RP/0/RSP0/CPU0:router(admin)# install upgrade source tftp://10.10.10.10/packages pkg1,pkg2
```

pkg1 is an optional package and pkg2 is a SMU. Both are active versions, but inactive on the system. The **install upgrade source** command checks whether the package or SMU is already inactive on system. If it is in inactive, the command skips its downloading, and adds as well as activates optional packages or SMUs along with its pre requisites.

# install remove

To delete inactive packages from a storage device, use the **install remove** command in EXEC or mode.

Administration EXEC Mode:
**install remove** {**id** *add-id* | *device***:***package* | **inactive**} [**sdr** *sdr-name*] [**prompt-level** {**default** | **none**}] [**asynchronous** | **synchronous**] [**test**]
EXEC Mode:
**install remove** {*device***:***package* | **inactive**} [**prompt-level** {**default** | **none**}] [**asynchronous** | **synchronous**] [**test**]

## Syntax Description

| | |
|---|---|
| **id** *add-id* | Specifies the ID number of an **install add** operation. The command deletes all packages that were added in the specified **install add** operation. The ID number of an **install add** operation is indicated in the syslog displayed during the operation and in the output of the **show install log** command.<br>Up to 16 **install add** operations can be specified |
| *device* **:** *package* | Device and package, expressed in concatenated form (for example, disk0:asr9k-mgbl-3.8.0). For the *device* argument, the value is a specified storage device, typically **disk0:**.<br>**Note**<br>Multiple packages can be removed at the same time. Up to 32 *device* **:** *package* pairs can be specified. |
| **inactive** | Removes all inactive, noncommitted packages from the boot device (usually disk0:). |
| **sdr** *sdr-name* | (Optional. Administration EXEC mode only) Removes a package for a specific secure domain router (SDR). The *sdr-name* argument is the name assigned to the SDR. |
| **prompt-level** {**default** \| **none**} | (Optional) Specifies when you are prompted for input during the procedure.<br>• **default** —You are prompted only when input is required by the operation.<br>• **none** —You are never prompted. |
| **asynchronous** | (Optional) Performs the command in asynchronous mode. In asynchronous mode, this command runs in the background, and the EXEC prompt is returned as soon as possible. This is the default mode. |
| **synchronous** | (Optional) Performs the command in synchronous mode. This mode allows the installation process to finish before the prompt is returned. |
| **test** | (Optional) Verifies the effects of proposed operations without making changes to the Cisco IOS XR software. |

## Command Default

The operation is performed in asynchronous mode: The **install remove** command runs in the background, and the EXEC prompt is returned as soon as possible.

**Command Modes** EXEC

**Command History**

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |
| Release 4.0.0 | This command was removed from EXEC mode.<br>Support was removed for the **sdr** keyword. |

**Usage Guidelines**

**Note** Only inactive packages can be removed. (Packages cannot be in the active or committed software set.)

- To remove all inactive packages from the boot device (usually **disk0:**), use the **install remove** command with the **inactive** keyword.
- To remove a specific inactive package from a storage device, use the **install remove** command with the *device: package* arguments.

**Note** When removing all inactive packages from the boot device, use the **show version**, **show install active**, or **show install committed** command to determine the device used as the boot device.

- To remove all packages that were added in one or more specific **install add** operations, use the **id** *add-id* keyword and argument. The operation ID of an **install add** operation is indicated in the syslog displayed during the operation and in the output of the **show install log** command. If you specify packages according to operation ID, all the packages that were added by the specified operation must still be on the router.

**Command Modes**

- To remove packages from the Owner SDR, use the **install remove** command in administration EXEC or EXEC mode.
- To remove all inactive packages from the boot device in the system or SDR, use the **install remove** command with the **inactive** keyword.

### User Prompts

Use the **install remove** command with the **prompt-level none** keywords to automatically ignore any confirmation prompts and proceed with the package removal.

### Test Operation

Use the **test** keyword to verify the effects of the package removal operation and determine whether the operation can be completed. After previewing the effects of the proposed operations, use the **show install log** command for more details about the effects of the proposed operations.



**Note** When removing a package, note that the **install remove** command ignores secure domain router (SDR) boundaries and performs the operation in global scope.

The following example shows how to remove a specific inactive package. In this example, the operation is run in test mode. The operation is then confirmed and the package is removed.

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# install remove
disk0:asr9k-diags-3.7.90 test

Install operation 30 'install remove disk0:asr9k-diags-3.7.90 test' started by user 'user_b'
 at 23:40:22 UTC Sat Apr 15 2006.
Warning:  No changes will occur due to 'test' option being specified. The
Warning:  following is the predicted output for this install command.
Info:     This operation will remove the following package:
Info:         disk0:asr9k-diags-3.7.90
Info:     After this install remove the following install rollback points will
Info:     no longer be reachable, as the required packages will not be present:
Info:         4, 9, 10, 14, 15, 17, 18
Proceed with removing these packages? [confirm] y

The install operation will continue asynchronously.
Install operation 30 completed successfully at 23.
```

The following example shows how to remove all inactive packages from the boot device:

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# install remove inactive synchronous

RP/0/RSP0/CPU0:Aug 15 09:25:41.020 :
   instdir[198]: %INSTALL-INSTMGR-6-INSTALL_OPERATION_STARTED :
Install operation 8 '(admin) install remove inactive' started by user 'user_b'
Install operation 8 '(admin) install remove inactive' started by user 'user_b' at
   09:25:41 UTC Tue Aug 15 2006.
Info:     This operation will remove the following package:
Info:         disk0:asr9k-compmgmt__installmgr-0.0.5
Proceed with removing these packages? [confirm]
The install operation will continue asynchronously.
```

# install replace

To replace the currently installed software with that in a given ISO and apply the change, use the **install replace** command in EXEC mode.

**install replace** *location* [**commit**] [**reload**] [**noprompt**] [**synchronous**]

| | |
|---|---|
| *location* | Specifes the location of the package for installation. |
| **commit** | (Optional) Commits the installed software after replacing. |
| **reload** | (Optional) Replaces the software through a reload. |
| **noprompt** | (Optional) Applies the changes without prompting for permission. |
| **synchronous** | (Optional) Applies the changes synchronously. |

## Command Default

None

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 6.5.2 | This command was introduced. |
| Release 25.4.1 | The **ftp and tftp** keywords are deprecated. |

## Usage Guidelines

Include the keyword **noprompt** in the command to enable the system to bypass your permission to reload the router.

When a router reloads using **install replace**, if the line card experiences a timeout waiting for the sysdb configuration replay, the system will notify the user once to reload the line card. This reload allows the sysdb configuration to come up properly after the line card restarts.

Starting with Cisco IOS XR Software Release 25.4.1, do not use the FTP or TFTP protocols for file transfers, as these are unsecure and deprecated. Instead, use SFTP (Secure File Transfer Protocol) or SCP (Secure Copy Protocol) for improved security, as they provide encrypted communication.

Syslog warnings will be displayed each time these commands are executed.

## Task ID

| Task ID | Operations |
|---|---|
| pkg-mgmt | read, write |

This example shows how to replace the current software with the asr9k-x64.iso image.

```
Router# install replace /harddisk:/asr9k-x64.iso
```

This example shows how to replace the current software and commit the changes:

```
Router# install replace /harddisk:/asr9k.iso commit
```

This example shows how to replace the current software and reload:

```
Router# install replace /harddisk:/asr9k.iso reload
```

# reload

**reload**

**Syntax Description** This command has no keywords or arguments.

**Command Default** No default behavior or values

**Command Modes** EXEC

**Command History**

| Releases | Modifications |
|---|---|
| Release 3.7.2 | This command was introduced. |

**Usage Guidelines** To use this command, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Use the **reload** command to cause the RSP to reload the Cisco IOS XR software according to the configuration register setting (for example, 0x0 to enter ROMMON mode and 0x2 to reload the RSP to EXEC mode). If a standby RSP is in the ready redundancy state, the **reload** command also causes the router to fail over to the standby RSP. Use the **show redundancy** command in EXEC mode to display the status of the standby RSP.

When the **reload** command is used and a switchover occurs, the running (active) software configuration is automatically maintained during switchover.



**Caution** If a standby RSP is not installed or is not in the ready state, then the router experiences a loss of service while the active RSP is reloading Cisco IOS XR software. To view the status of the standby RSP, issue the **show redundancy** command in EXEC mode.

If you use the **reload** command and there is no available standby node, you are prompted to continue with the reload:

When a router reloads using **reload**, if the line card experiences a timeout waiting for the sysdb configuration replay, the system will notify the user once to reload the line card. This reload allows the sysdb configuration to come up properly after the line card restarts.

```
RP/0/RSP0/CPU0:router# reload

Standby card not present or not Ready for failover. Proceed?[confirm]y
```

**Task ID**

| Task ID | Operations |
|---|---|
| root-lr | execute |

The following example shows how to reload the active RSP. If a standby RSP is in the ready state, then the router fails over to the standby RSP. If the standby RSP is not installed or is not in the ready state, then the router enters ROMMON mode and routing operations stop.

```
RP/0/RSP0/CPU0:router# reload

  Updating Commit Database.  Please wait...[OK]
  Proceed with reload? [confirm] y

  PCI0 device[7]: Vendor ID 0x10ee
  PCI0 device[7]: Device ID 0x300e
  PCI1 device[7]: Device ID 0x1100
  PCI1 device[7]: Vendor ID 0x1013
  PCI1 device[8]: Device ID 0x649
  PCI1 device[8]: Vendor ID 0x1095
  PCI1 device[9]: Device ID 0x5618
  PCI1 device[9]: Vendor ID 0x14e4
  PCI1 device[10]: Device ID 0x5618
  PCI1 device[10]: Vendor ID 0x14e4
  System Bootstrap, Version 1.15(20040120:002852) ,
  Copyright (c) 1994-2004 by cisco Systems, Inc.
  Board type is 0x100000 (1048576)
  Enabling watchdog
  Broadcom 5618 #0 Found on PCI
  Broadcom 5618 #1 Found on PCI
  No. of BCM 56xx switches found 2 .
  BCM Switch #0 initialisation complete.
  BCM Switch #1 initialisation complete
  G4(7450-SMP-GT64260_A) platform with 2048 Mb of main memory

  rommon B1 >
```

# ztp enable

Manual Zero Touch Provisioning (ZTP) invocation using CLI commands allows ZTP to run over more interfaces.

To enable Zero Touch Provisioning (ZTP) at boot, use the **ztp enable** command in EXEC mode.

**ztp enable**

## Command Default

No default behavior or values

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

By default, ZTP is enabled. When you execute the **ztp enable** command the start value in the `ztp.ini` file is set to *True*.

The following example shows the sample of the `ztp.ini` file:

```
[Startup]
start: True
retry_forever: True

[Fetcher Priority]
Mgmt4: 0
Mgmt6: 1
DPort4: 2
DPort6: 3
```

This example shows how to enable ztp at boot:

```
Router#ztp enable
Fri Jul 12 16:09:02.154 UTC
Enable ZTP? [confirm] [y/n] :y
ZTP Enabled.
```

# ztp disable

Manual Zero Touch Provisioning (ZTP) invocation using CLI commands allows ZTP to run over more interfaces.

To disable Zero Touch Provisioning (ZTP) at boot, use the **ztp disable** command in EXEC mode.

**ztp disable**

## Command Default

No default behavior or values

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

Use **ztp disable** command to disable ZTP. When you execute the **ztp disable** command, the start value in the `ztp.ini` file is set to *False*.

The following example shows the sample of the `ztp.ini` file:

```
[Startup]
start: False
retry_forever: True

[Fetcher Priority]
Mgmt4: 0
Mgmt6: 1
DPort4: 2
DPort6: 3
```

This example shows how to disable ztp at boot:

```
Router#ztp disable
Fri Jul 12 16:07:18.491 UTC
Disable ZTP? [confirm] [y/n] :y
ZTP Disabled.
Run ZTP enable to run ZTP again.
```

# ztp initiate

To invoke a new ZTP DHCP session, use the **ztp initiate** command in EXEC mode.

**ztp initiate {[ apply** *configuration* **]** | **[ dataport ]** | **[ dhcp4 ]** | **[ dhcp6 ]** | **[ dhcp4-client-identifier ]** | **[ dhcp6-client-identifier ]** | **[ dscp** *value* **]** | **[ dscp6** *value* **]** | **[ hostname ]** | **[ interface ]** | **[ management ]** | **[ noprompt ]} [ debug ] [ verbose ]**

## Syntax Description

| | |
|---|---|
| **debug** | Run with additional logging to the console(cisco-support) |
| **verbose** | Run with logging to the console(cisco-support) |
| **apply** *configuration* | XR configuration commands to apply(cisco-support) |
| **dataport** | Send DHCP requests on all ADMIN UP physical LC interfaces. |
| **dhcp4** | Send only DHCP IPv4 requests(cisco-support) |
| **dhcp6** | Send only DHCP IPv6 requests(cisco-support) |
| **dhcp4-client-identifier** | Override default dhcp-client-identifier(cisco-support) |
| **dhcp6-client-identifier** | Override default dhcp6-client-id(cisco-support) |
| **dscp** *value* | DSCP/Prec Value(cisco-support) |
| **dscp6** *value* | DSCP6/Prec Value(cisco-support) |
| **hostname** | XR hostname to set(cisco-support) |
| **interface** | Send DHCP requests only on the given interface(cisco-support) |
| **management** | Send DHCP requests on the platforms management interface(cisco-support) |
| **noprompt** | Run without prompting(cisco-support) |

## Command Default

No default behavior or values

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 6.3.1 | This command was introduced. |

## Usage Guidelines

Use the **ztp initiate** command to forceably inititate the ZTP, ignoring username configuration. **ztp initiate** allows the execution of a script even when the system has already been configured. This command is useful for testing ZTP without forcing a reload. This command is particularly useful to test scripts or if some manual operations are required before provisioning the box. **ztp initiate** can specify any data interfaces and management interface on the system to be used for the whole ZTP process. If you don't specify an interface, ztp will be invoked on management interface only.

No progress logs are shown by default, although there will be XR syslogs for important events. If you wish to see more logs, add **verbose** after the **ztp initiate** command. For more details, add **debug** before **verbose**.

Logs can be found in **disk0:/ztp/ztp.log**.

### Example

This example shows how to bring up the interface manually:

```
RP/0/RP0/CPU0:router#ztp initiate debug verbose interface TenGigE 0/0/0/0
Invoke ZTP? (this may change your configuration) [confirm] [y/n] :
```

This example shows how to get rid of the prompting:

```
RP/0/RP0/CPU0:router#ztp initiate noprompt
Mon Jun 27 20:40:10.353 UTC
ZTP will now run in the background.
Please use "show logging" or look at /disk0:/ztp/ztp.log to check progress.
```

This example shows how to invoke the breakout discovery and ZTP, ZTP is invoked on the interfaces which are up:

```
RP/0/RP0/CPU0:router#ztp breakout debug verbose
RP/0/RP0/CPU0:router#ztp initiate dataport debug verbose
Invoke ZTP? (this may change your configuration) [confirm] [y/n] :
```