



# Configuration Management Commands

---

- [abort](#), on page 3
- [alias](#), on page 4
- [apply-template](#), on page 7
- [clear comment](#), on page 8
- [clear configuration commits](#), on page 9
- [clear configuration inconsistency](#), on page 11
- [clear configuration inconsistency replica](#), on page 13
- [clear configuration sessions](#), on page 14
- [commit](#), on page 15
- [configuration commit auto-save](#), on page 19
- [configure](#), on page 21
- [description \(interface\)](#), on page 23
- [do](#), on page 24
- [end](#), on page 25
- [end-group](#), on page 27
- [end-template](#), on page 28
- [exclude-group](#), on page 29
- [exit](#), on page 30
- [hostname](#), on page 32
- [load](#), on page 33
- [load commit changes](#), on page 34
- [load configuration failed](#), on page 35
- [load configuration removed](#), on page 36
- [load rollback changes](#), on page 37
- [man](#), on page 38
- [more](#), on page 41
- [pwd \(config\)](#), on page 45
- [rollback configuration](#), on page 46
- [root](#), on page 49
- [save configuration](#), on page 51
- [save configuration changes](#), on page 52
- [save configuration commit changes](#), on page 53
- [save configuration failed](#), on page 54

- [save configuration merge](#), on page 55
- [save configuration removed](#), on page 56
- [save rollback changes](#), on page 57
- [service cli commit-optimized enable](#), on page 58
- [set default-afi](#), on page 59
- [set default-safi](#), on page 60
- [set default-vrf](#), on page 61
- [show](#), on page 63
- [show aliases](#), on page 66
- [show apply-group](#), on page 67
- [show configuration failed](#), on page 68
- [show configuration changes](#), on page 69
- [show configuration commit changes](#), on page 71
- [show configuration commit list](#), on page 75
- [show configuration \(config\)](#), on page 77
- [show configuration failed incompatible](#), on page 79
- [show configuration failed remove](#), on page 80
- [show configuration failed rollback](#), on page 82
- [show configuration failed startup](#), on page 83
- [show configuration history](#), on page 84
- [show configuration inconsistency replica](#), on page 87
- [show configuration persistent](#), on page 88
- [show configuration removed](#), on page 90
- [show configuration rollback changes](#), on page 91
- [show configuration running](#), on page 93
- [show configuration running-config](#), on page 94
- [show configuration sessions](#), on page 96
- [show default-afi-safi-vrf](#), on page 98
- [show history](#), on page 99
- [template](#), on page 101

# abort

To terminate a configuration session and discard all uncommitted changes without system confirmations, use the **abort** command in any configuration mode.

## abort

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

**Command Default** None

**Command Modes** Any configuration mode

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the **abort** command to terminate a configuration session and return to EXEC mode from any configuration mode. This command discards all uncommitted configuration changes. You are prompted to commit the changes.

Task ID	Task ID	Operations
	Task ID for the feature or mode impacted by the command	Operation for the feature or mode impacted by the command

The following example shows how to use the **abort** command to discard all changes made during a configuration session:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)#
RP/0/RP0/CPU0:router(config-if)# ipv4 address 1.1.1.1 255.0.0.0
RP/0/RP0/CPU0:router(config-if)# abort
RP/0/RP0/CPU0:router#
```

# alias

To create a command alias, use the **alias** command in XR Config mode. To delete an alias, use the **no** form of this command.

```
alias alias-name[(param-list)]content
no alias alias-name
```

## Syntax Description

<i>alias-name</i>	Name of the command alias. Alias names can be a single word or multiple words joined by a hyphen (-) or an underscore (_).
<i>param-list</i>	(Optional) Parameters assigned to the alias. These parameters are filled in at execution time.
<i>content</i>	Original command syntax. Valid abbreviations of the original command syntax can be entered for the <i>content</i> argument.

## Command Default

No command aliases are configured.

## Command Modes

XR Config mode

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

Cisco IOS XR software supports generic alias definitions for various entities. Any physical or logical entity can have an alias as a reference. For example, an alias can refer to a command, a partial command, a group of commands, a location, or an IP address.

An alias must first be defined. The alias can then be used in command lines in place of the defined entity.

Following is a list of properties for an alias:

- An alias can be used anywhere and in any mode.
- An alias can have zero, one, or many parameters.
- An alias can refer to those parameters with the \$ sign.
- If an alias refers to more than one command, the commands must be separated by a semicolon (;).
- The size of the **alias** command is limited to 1024 characters.

The alias command can be used anywhere. If the content referenced by the alias is invalid or inappropriate in that context or mode, the system issues a warning message containing the substituted content.

An alias name should not be a subset of the keywords that it represents as alias. Substitution is done only when the entered input match fails completely. For instance, the attempt to define an alias with “config?” as the alias name fails, as shown in the following example:

```
RP/0/RP0/CPU0:router(config)# alias config set_host hostname router
RP/0/RP0/CPU0:router(config)# show configuration

alias set_host hostname router
```

Use the **show aliases** command to display all command aliases or the command aliases in a specified mode.

Task ID	Task ID	Operations
	logging	read, write

The following example shows how to create an alias named `ipbr` for the `show ipv4 interface brief` command, commit the configuration, enter XR EXEC mode and then enter the configured alias:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# alias ipbr show ipv4 interface brief
RP/0/RP0/CPU0:router(config)# show configuration

Building configuration...
alias ipbr show ipv4 interface brief
end
RP/0/RP0/CPU0:router(config)# commit
RP/0/RSP0/CPU0:Feb 21 04:42:57.017 : config[65689]: %MGBL-LIBTARCFG-6-COMMIT :
Configuration committed by user 'lab'. Use 'show configuration commit changes 1000000022'
to view the changes.
RP/0/RSP0/CPU0:router(config)# end
RP/0/RP0/CPU0:Mar 27 22:19:05 : config[65739]: %SYS-5-CONFIG_I : Configured from console
by lab
RP/0/RP0/CPU0:router# ipbr
RP/0/RP0/CPU0:router# show ipv4 interface brief

Interface                               IP-Address      Status           Protocol
-----                               -
Loopback0                               1.1.1.1         Up               Up
Loopback999                             unassigned      Up               Up
MgmtEth0/0/CPU0/0                       12.29.56.21    Up               Up
RP/0/RP0/CPU0:router#
```

The following example shows how to define an alias, `mycompany-10ge`, for POS interface `1/0/2/3` and then how to use that alias to shut down the interface:

```
RP/0/RP0/CPU0:router(config)# alias mycompany-10ge gigabitethernet1/0/2/3
RP/0/RP0/CPU0:router(config)# interface mycompany-10ge
RP/0/RP0/CPU0:router(config-if)# shutdown
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)#
```

The following example shows the use of a parameter name in an alias definition:

```
RP/0/RP0/CPU0:router(config)# alias shint (intname) show interface $intname
```

The following example shows an alias defined with one parameter and two commands:

```
RP/0/RP0/CPU0:router(config)# alias shint_both (intname) show interface $intname;show run
interface $intname
```

The following example shows the use of the alias `shint_both` in EXEC mode:

```
RP/0/RP0/CPU0:router(exec)# shint_both(gigabitethernet1/2/3/4)
```

Two commands are issued, as follows:

```
RP/0/RP0/CPU0:router(exec)# show interface gigabitethernet1/2/3/4; show run interface  
gigabitethernet1/2/3/4
```

# apply-template

To apply a template to the target configuration, use the **apply-template** command in XR Config mode.

**apply-template** *template-name* [(*param-list*)]

<b>Syntax Description</b>	<i>template-name</i> Name of the template to be applied to the running configuration. Use the <b>template</b> command to define a template.
---------------------------	---

<i>param-list</i>	(Optional) Up to five template parameters.
-------------------	--

<b>Command Default</b>	No templates are applied to the target configuration.
------------------------	---

<b>Command Modes</b>	XR Config mode
----------------------	----------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines**

Use the **apply-template** command to apply a template to the target configuration. Templates allow you to create a template name that represents a group of configuration commands.

Use the **template** command to define a template. Use the **end-template** command to exit template configuration mode and return to global configuration mode. Use the **show-running** command with the optional **template** *template-name* keyword and argument to display the contents of a template.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	read, write

The following example shows how to define a template and then apply the template to the target configuration:

```
RP/0/RP0/CPU0:router(config)# template hostname-template
RP/0/RP0/CPU0:router(config-TPL)# hostname router1
RP/0/RP0/CPU0:router(config-TPL)# end-template
RP/0/RP0/CPU0:router(config)# apply-template hostname-template
```

# clear comment

To discard a comment associated with a configuration, use the **clear comment** command in any configuration or XR Config mode.

## clear comment

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

**Command Default** None

**Command Modes** Any configuration mode  
XR Config mode

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** The **clear comment** command clears any comments that were added for a specific configuration in the configuration file. After you enter the **clear comment** command, enter the configuration for which you want to delete the comment on a separate line.

To enter configuration comments, enter ! followed by the comment. The comment you enter is associated with the next configuration entered. For example:

```
RP/0/RP0/CPU0:router#!router1 is located in xxx
RP/0/RP0/CPU0:router# hostname router1
RP/0/RP0/CPU0:router# commit
```

The comment is displayed in the output of the **show running-config** command:

```
RP/0/RP0/CPU0:router# show running-config
...
!router1 is located in xxx
hostname router1
...
```

Task ID	Task ID	Operations
	Task ID for the feature or configuration mode impacted by the command	Operation for the feature or configuration mode impacted by the command

The following example shows how to discard the comment associated with the configuration ipv4 address 1.1.1.1 255.0.0.0.

```
RP/0/RP0/CPU0:router(config-if)# clear comment
RP/0/RP0/CPU0:router(config-if)# ipv4 address 1.1.1.1 255.0.0.0
```



# clear configuration commits

To delete old commit IDs from the commit database to free up disk space, use the **clear configuration commits** command in or XR EXEC mode.

**clear configuration commits** {**diskspace** *kilobytes* | **oldest** *number-of-commits*}

<b>Syntax Description</b>	<b>diskspace</b> <i>kilobytes</i>	Deletes as many commit IDs (beginning with the oldest available commit ID) from the commit database as required to free the number of kilobytes (KB) specified for the <i>kilobytes</i> argument. The range for the number of kilobytes of disk space to free is 1 to 4194304.
		<b>Note</b> The amount of disk space freed may vary depending on the size and number of commits present in the commit database.
	<b>oldest</b> <i>number-of-commits</i>	Deletes the number of commit IDs specified for the <i>number-of-commits</i> argument.
		<b>Note</b> Use the online help ( ? ) function to display the range of commit IDs available for deletion.

**Command Default** None

**Command Modes** XR EXEC mode

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the **clear configuration commits** command to delete the number of commit IDs available for rollback operations. The most recent 100 commits are retained by the system. As new commit IDs are added, the oldest commit IDs are discarded and are no longer available for rollback operations.



**Note** The **clear configuration commits** command deletes commits from the commit database only. The running configuration, thus, is not changed.



**Note** When a commit ID is deleted from the commit database, it is no longer available for rollback and can no longer be used to display commit changes (with the **show configuration rollback changes** command).

Use the **rollback configuration** command to roll back the current running configuration to a previous configuration. Use the **show configuration rollback changes** command to display a list of the commit IDs available for rollback operations or to display the changes that would be made by the **rollback configuration** command.

Task ID	Task ID	Operations
	config-services	execute

The following example shows how to delete the oldest 16 commit IDs to free up disk space. After entering this command, you will be prompted to confirm the deletion.

```
RP/0/RP0/CPU0:router# clear configuration commits oldest 16
```

```
Deleting 16 rollback points '1000000021' to '1000000036'  
256 KB of disk space will be freed. Continue with deletion?[confirm] y
```

# clear configuration inconsistency

To clear an inconsistency alarm for a router configuration, use the **clear configuration inconsistency** command in XR EXEC mode.

**clear configuration inconsistency**

## Syntax Description

This command has no keywords or arguments.

## Command Default

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

## Command Modes

XR EXEC mode

## Command History

Release	Modification
Release 3.0.0	This command was introduced.

## Usage Guidelines

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 processor (RP) card is inserted or when there is an OIR (Online Insertion and Removal).

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.

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 router, enter the **clear configuration inconsistency** command in XR EXEC mode.

## Task ID

Task ID	Operations
config-services	execute

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

```
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
```

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

```
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

# clear configuration inconsistency replica

To resolve configuration inconsistencies on a replica node, use the **clear configuration inconsistency replica** command in administration EXEC or XR EXEC mode.

**clear configuration inconsistency replica location** *node-id*

<b>Syntax Description</b>	<b>location</b> <i>node-id</i> Resolves the configuration inconsistencies on the designated node. The <i>node-id</i> argument is expressed in the <i>rack/slot</i> notation.
---------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>Command Default</b>	Administration EXEC mode: Resolves any configuration inconsistencies for the admin plane configuration. XR EXEC mode: Resolves any configuration inconsistencies for the configuration.
------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>Command Modes</b>	XR EXEC mode
----------------------	--------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Usage Guidelines</b>	In administration EXEC mode, the replica node for the <b>clear configuration inconsistency replica</b> command is the standby designated system controller (DSC). In EXEC mode, the replica nodes are the route switch processors (RSPs) that can become the designated shelf controller (DSC).
-------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Use the **clear configuration inconsistency replica** command if there is a configuration inconsistency between the standby DSC and the current active DSC; or alternatively, if the configuration on any nodes that could become the DSC is not the same as the configuration on the current DSC. To determine if you have a configuration inconsistency, use the **show configuration inconsistency replica** command.

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

To clear configuration inconsistencies for an SDR configuration, enter the **clear configuration inconsistency replica** command in EXEC mode for that SDR.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	execute

The following example shows how to clear any configuration inconsistencies for the DSC configuration by using the **clear configuration inconsistency replica** command in EXEC mode:

```
RP/0/RP0/CPU0:router# clear configuration inconsistency replica location 0/rp1/cpu0
```

The replica has been repaired.

# clear configuration sessions

To clear (end) an active configuration session, use the **clear configuration sessions** command in administration EXEC or XR EXEC mode.

**clear configuration sessions** *session-id*

<b>Syntax Description</b>	<i>session-id</i> Identifier for the configuration session to be terminated.
---------------------------	------------------------------------------------------------------------------

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	Administration EXEC XR EXEC mode
----------------------	-------------------------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Usage Guidelines</b>	Use the <b>clear configuration sessions</b> command to clear a configuration session. This command can be used to end the configuration sessions of another user. Any uncommitted changes to a user's target configuration are discarded.
-------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Use the **show configuration sessions** command to identify active configuration sessions.

When a configuration session is cleared, a message is displayed on the terminal of the terminated user. For example:

```
RP/0/RP0/CPU0:router(config)# This configuration session was terminated by user 'user_a'
from line 'aux0_0_CPU0'
```

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	execute

The following example shows how to clear an active configuration session. In this example, the **show configuration sessions** command displays the active configuration session. The **clear configuration sessions** command clears the active configuration session.

```
RP/0/RP0/CPU0:router# show configuration sessions

Current Configuration Session  Line      User      Date                               Lock
00000211-002c409b-00000000    con0_1_CPU0  UNKNOWN  Mon Feb  2 01:02:09 2009

RP/0/RP0/CPU0:router# clear configuration sessions 00000211-002c409b-00000000

session ID '00000211-002cb09b-00000000' terminated
```

# commit

To commit the target configuration to the active (running) configuration, use the **commit** command in any configuration XR Config mode System Admin Config mode.

**commit** [**best-effort**] [**comment** *line*] [**confirmed** [{*seconds* | **minutes** *minutes*}] ] [**force**] [**label** *line*] [**replace**] [**save-running filename** *file\_path*]

Syntax Description		
<b>best-effort</b>		(Optional) Merges the target configuration with the running configuration and commits only valid changes (best effort). Some configuration changes might fail due to semantic errors.
<b>comment</b> <i>line</i>		(Optional) Assigns a comment to a commit. This text comment is displayed in the commit entry displayed in the output for the <b>show configuration commit list</b> command with the optional <b>detail</b> keyword.
<b>confirmed</b> [ <i>seconds</i>   <b>minutes</b> <i>minutes</i> ]		(Optional) Commits the configuration on a trial basis for the time specified in seconds or minutes.  <b>Note</b> The <b>confirmed</b> option is not available in administration configuration mode.
<b>force</b>		(Optional) Forces a commit operation in low-memory conditions.
<b>label</b> <i>line</i>		(Optional) Assigns a meaningful label. This label is displayed (instead of the autogenerated commit ID) in the output for the <b>show configuration commit list</b> .
<b>replace</b>		(Optional) Replaces the entire running configuration with the contents of the target configuration.
<b>save-running filename</b> <i>file_path</i>		(Optional) Saves the running configuration to a specified file.

**Command Default** The default behavior is *pseudo-atomic*, meaning that all changes must succeed for the entire commit operation to succeed. If any errors are found, none of the configuration changes take effect.

**Command Modes** Any configuration mode  
XR Config mode  
System Admin Config mode

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines**

Changes made during a configuration session are inactive until the **commit** command is entered. By default, the commit operation is *pseudo-atomic*, meaning that all changes must succeed for the entire commit operation to succeed. If any errors are found, none of the configuration changes takes effect.

To replace the default numeric ID for the commit, use the optional **label** keyword. This label is displayed (instead of the autogenerated commit ID) in the output for the **show configuration commit list** command.

Enter an optional comment with the **comment** keyword to provide additional information about the commit action. This comment is displayed in the output for the **show configuration commit list** command with the **detail** keyword.

Use the optional **confirmed minutes** keyword and argument to commit a configuration on a trial basis for a minimum of 30 seconds and a maximum of 300 seconds (5 minutes). During the trial configuration period, enter the **commit** command to confirm the configuration. If the **commit** command is not entered, then the system reverts to the previous configuration when the trial time period expires. The confirmed option is not available in administration configuration mode.

You can use the **commit** command in conjunction with the **load** command. Load a new configuration with the **load** command, and use the **commit** command with the **replace** keyword to have the loaded configuration become the active (running) configuration.

Use the optional **save-running filename file\_path** keywords and argument to save the running configuration to a specified file. To configure automatic saving of the configuration file on every commit, use the **configuration commit auto-save** command. If automatic saving of the configuration file is already enabled, specifying **save-running filename file\_path** with the **commit** command has no additional effect.

In pseudo-atomic commit, if an error occurs on one or more of the configurations in a commit, other configurations which are already part of the running configuration in the same commit are reverted.



**Caution** Saving the running configuration to a file is CPU intensive.



**Note** If you use the **commit** command without previously loading a target configuration, a blank configuration is committed.



**Note** If you use the **commit** command with the **replace** keyword, it does not affect the mode of an 8-port E1/T1 SPA. If the mode is E1 before using the **commit replace** command, it remains E1. However, since the default mode is T1, the router does not recognize that the mode is E1. To change the mode to T1, you must first use the **hw-module subslot cardtype e1** command to add the E1 mode into the configuration so that it correlates with the system. Then manually reload the router and it boots in T1 mode.

For more information regarding the **hw-module subslot cardtype** command, refer to *Interface and Hardware Component Command Reference for Cisco 8000 Series Routers*.

**Task ID****Task ID****Operations**

Task ID for the feature or configuration mode impacted by the command

Operation for the feature or configuration mode impacted by the command



### Committing the Target Configuration to the Active Running Configuration

The following example shows how to commit the target configuration to the active running configuration. In this example, the **commit** command saves changes to the router hostname.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# hostname router1
RP/0/RP0/CPU0:router(config)# commit

RP/0/RP0/CPU0:Feb 21 04:42:57.017 : config[65689]: %MGBL-LIBTARCFG-6-COMMIT :
Configuration committed by user 'user_a'.
Use 'show configuration commit changes 1000000033' to view the changes.
```

### Adding a Comment to a Configuration Commit

The following example shows how to use the **commit** command with the optional **comment line** keyword and argument to assign a text description to the commit operation. The comment is then displayed in the output of the **show configuration commit list** command with the **detail** keyword.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# hostname router2
RP/0/RP0/CPU0:router(config)# commit comment new name for router

RP/0/RP0/CPU0:Feb 21 04:42:57.017 : config[65689]: %MGBL-LIBTARCFG-6-COMMIT :
Configuration committed by user 'user_a'. Use 'show configuration commit
changes 1000000226' to view the changes.

RP/0/RP0/CPU0:router2(config)# end
RP/0/RP0/CPU0:router2# show configuration commit list detail

1) CommitId: 1000000226                Label: NONE
   UserId:   user_a                    Line:   con0_RP1_CPU0
   Client:   CLI                       Time:   12:59:26 UTC Wed Feb 04 2004
   Comment:  new name for router

2) CommitId: 1000000225                Label: NONE
   UserId:   user_a                    Line:   con0_RP1_CPU0
   Client:   CLI                       Time:   12:58:32 UTC Wed Feb 04 2004
   Comment:  NONE
```

### Changing the Commit ID to a Text Label

The following example shows how to use the **commit** command with the optional **label line** keyword and argument to change the commit ID to a text label for easier identification. The label is then displayed in the output of the **show configuration commit list** command.

```
RP/0/RP0/CPU0:router2# configure
RP/0/RP0/CPU0:router2(config)# hostname router3
RP/0/RP0/CPU0:router2(config)# commit label new_name

RP/0/RP0/CPU0:Feb 21 04:42:57.017 : config[65689]: %MGBL-LIBTARCFG-6-COMMIT :
Configuration committed by user 'user_a'.
Use 'show configuration commit changes 1000000227' to view the changes.
```

```
RP/0/RP0/CPU0:router3(config)# end
RP/0/RP0/CPU0:router3# show configuration commit list

SNo. Label/ID      User      Line      Client      Time Stamp
~~~~ ~~~~~~      ~~~~~      ~~~~~      ~~~~~      ~~~~~~
1   new_name      user_a    con0_1_C  CLI         13:00:53 UTC Wed Feb 04 2004
2   1000000226    user_a    con0_1_C  CLI         12:59:26 UTC Wed Feb 04 2004
3   1000000225    user_a    con0_1_C  CLI         12:58:32 UTC Wed Feb 04 2004
```

### Commit a Configuration for a Specified Time

The following example shows how to use the **commit** command with the optional **confirmed** keyword and number *argument*. The configuration changes are committed only for the specified number of seconds. You can then either confirm the commit operation or discard the changes.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# hostname router3
RP/0/RP0/CPU0:router(config)# commit confirmed 30
RP/0/RP0/CPU0:router3(config)# end
```

## configuration commit auto-save

To enable automatic saving of the running configuration to a specified file on every commit, use the **configuration commit auto-save** command in XR Config mode. To disable automatic saving of the running configuration to a specified file on every commit, use the **no** form of the command.

```
configuration commit auto-save [{ filename file_path | password { clear { encryption-aes
| encryption-default } password | encrypted | { encryption-aes | encryption-default } password } |
maximum number | timestamp | wait-time duration }]
```

### Syntax Description

<b>filename</b> <i>file_path</i>	Specifies the location to which to save the running configuration.
<b>password</b> <i>password</i>	Specifies the password of the remote URL.
<b>clear</b>	Specifies an unencrypted password.
<b>encryption-aes</b>	Saves the password in aes encrypted form.
<b>encryption-default</b>	Saves password in default encrypted form.
<b>encrypted</b>	Specifies an encrypted password.
<b>maximum</b> <i>number</i>	Specifies how many maximum auto-saves are possible.
<b>timestamp</b>	Includes the timestamp in the auto-save filename.
<b>wait-time</b> <i>duration</i>	Specifies how long to wait to auto-save after the commit done to start the next auto-save. The <i>duration</i> can be specified as: days <i>days</i>   hours <i>hours</i>   minutes <i>minutes</i>   seconds <i>seconds</i>

### Command Default

None

### Command Modes

XR Config mode  
System Admin Config mode

### Command History

Release	Modification
Release 7.10.1	This command was modified to support public key authentication.
Release 7.9.1	This command was modified to include, <b>scp</b> , <b>sftp</b> , <b>password</b> , <b>wait-time</b> , <b>timestamp</b> , and <b>maximum</b> keywords.
Release 7.0.12	This command was introduced

### Usage Guidelines

The **configuration commit auto-save** command configures the system to save the running configuration to the specified file and location every time a **commit** command is run. Alternatively, you can save the configuration on a one-time basis by specifying the **save-running** keyword when you run the **commit** command. Use the following syntax when using **tftp**, **ftp**, **scp**, **sftp** or **rcp** as options.

From Cisco IOS XR Release 7.9.1 scp and sftp options are available under **filename**.

[ *tftp/ftp/rcp/scp/sftp* ]




---

**Caution** Saving the running configuration to a file is CPU intensive.

---

Task ID	Task ID	Operations
	config-services	write

The following example shows how to configure the system to save the running configuration to the file `//test-folder/test_123`, backup encrypted password, append time-stamp, configure maximum number of auto-save files possible, and specify wait-time before backing up the files whenever the configuration is committed:

```
Router#configure
Router(config)#configuration commit auto-save
Router(config-cfg-autosave)#filename sftp://user1@server1://test-folder/test_123
Router(config-cfg-autosave)#password clear encryption-default cisco
Router(config-cfg-autosave)#timestamp
Router(config-cfg-autosave)#maximum 10
Router(config-cfg-autosave)#wait-time days 0 hours 0 minutes 0 seconds 5
Router(config-cfg-autosave)#commit
```

While you are using public key authentication to save the running configuration, you don't need to mention password.

# configure

To enter global configuration mode or administration configuration mode, use the **configure** command in XR EXEC mode or .

**configure** [{**exclusive** | **terminal**}]

<b>Syntax Description</b>	<b>exclusive</b> (Optional) Locks the router configuration. The system configuration can be made only from the login terminal.				
	<b>terminal</b> (Optional) Configures the system from the login terminal. This is the default.				
<b>Command Default</b>	If the <b>configure</b> command is entered without a keyword, the system is configured from the login terminal.				
<b>Command Modes</b>	XR EXEC mode				
<b>Command History</b>	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>Release 7.0.12</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				

**Usage Guidelines**

Configuration modes are used to enter changes to a target configuration session and commit those changes to the running configuration. A router running Cisco IOS XR software contains multiple configurations:

- The configuration for a router. This mode is used to configure - specific features such as routing protocols.
- The administration configuration for system-wide resources and settings. Some features can be configured only in administration configuration mode.

## XR Config mode

Use the **configure** command in XR EXEC mode to enter XR Config mode and create a new target configuration for an SDR. From global configuration mode, you can enter any configuration mode. Configuration changes entered in global configuration mode impact the SDR to which the user is currently logged in.

## System Admin Config mode

Use the **configure** command in to enter System Admin Config mode and create a new target configuration. From , you can enter any configuration mode. Configuration changes entered in can impact resources for the entire router. See the command reference documentation for a specific command to determine the impact of commands entered in .

## Router Prompt

After you enter the **configure** command, the system appends “(config)” to the router prompt, indicating that the router is in a configuration mode. For example:

- The following prompt indicates that you are in global configuration mode for an SDR:

```
RP/0/RP0/CPU0:router(config)#
```

- The following prompt indicates that you are in administration configuration mode:

```
RP/0/RP0/CPU0:router(admin-config)#
```

### Locking a Configuration Session

To lock the configuration so that no other user can commit changes to the running configuration during your configuration session, issue the **configure** command with the **exclusive** keyword.

### Committing Changes and Returning to XR EXEC mode or

Changes to the target configuration remain inactive until the **commit** command is entered. To leave global configuration or administration configuration mode and return to the XR EXEC mode or prompt, issue the **end** or **exit** command; you are prompted to commit any uncommitted changes.

To leave configuration mode and return directly to XR EXEC mode or without being prompted to commit changes and without saving changes to the target configuration, enter the **abort** command in any configuration mode.

The following example shows how to enter global configuration mode from XR EXEC mode and then enter interface configuration mode to configure an IPv4 address, the **configure** command commits the configuration, and the **end** command terminates the configuration session and return the router to XR EXEC mode.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tengige 0/1/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 1.1.1.1 255.0.0.0
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:router(config-if)# end
RP/0/RP0/CPU0:router#
```

## description (interface)

To add a description to an interface configuration, use the **description** command in interface configuration mode. To remove the description, use the **no** form of this command.

**description** *comment*  
**no description**

### Syntax Description

*comment* Comment or a description applied to the interface. The maximum number of characters is 1022.

### Command Default

No description is configured.

### Command Modes

Interface configuration

### Command History

Release	Modification
Release 7.0.12	This command was introduced.

### Usage Guidelines

Use the **description** command to add a description to an interface configuration. The maximum number of characters is 1022.

### Task ID

Task ID	Operations
interface	read, write

The following example shows how to add a description to an interface configuration. In this example, the **description** command names a Management Ethernet interface.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface mgmteth 0/
1/CPU0/0
RP/0/RP0/CPU0:router(config-if)# description Management Ethernet Interface
```

# do

To execute an EXEC mode command from a configuration mode, use the **do** command in any configuration mode.

**do** *exec-command*

---

**Syntax Description**     *exec-command* EXEC mode command to be executed.

---

**Command Default**     None

**Command Modes**     Any configuration mode

---

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

---

**Usage Guidelines**     To display the various mode commands that are available to execute with the **do** command, use the online help (?) function at the configuration mode prompt.




---

**Note**     The **configure** and **describe** commands are not supported with the **do** command.

---



---

Task ID	Task ID	Operations
	Task ID for the EXEC command that you are using	read

---

The following example shows how to execute an EXEC command from interface configuration mode. In this example, the **do** command displays output from the **show protocols** command within interface configuration mode:

```
RP/0/RP0/CPU0:router(config)# interface tengige 0/1/0/1
RP/0/RP0/CPU0:router(config-if)# do show protocols

Routing Protocol "BGP 1"

Address Family IPv4 Unicast:
  Distance: external 20 internal 200 local 200
```



# end

To terminate a configuration session and return directly to XR EXEC mode , use the **end** command in any configuration mode.

**end**

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

**Command Default** None

**Command Modes** Any configuration mode

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the **end** command to exit any configuration mode and return directly to XR EXEC mode . If you enter this command without committing the changes to the target configuration, you are prompted to do so:

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode .

If errors are found in the running configuration, the configuration session does not end. To view the errors, enter the **show configuration** (config) command with the **failed** keyword.

- Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.



**Note** Entering **Ctrl-Z** is functionally equivalent to entering the **end** command.

Use the **abort** command to exit the configuration session and return to XR EXEC mode without being prompted to commit changes and without saving changes to the target configuration.

Task ID	Task ID	Operations
	config-services	read, write

The following example shows how to use the **end** command to end a configuration session. Changes stored in the target configuration are committed by answering **yes**.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tengige 0/2/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 1.1.1.1 255.0.0.0
```

**end**

```
RP/0/RP0/CPU0:router(config-if)# end
```

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: yes
```

```
RP/0/RP0/CPU0:router#
```

# end-group

To exit from configuration group submode and return to global configuration mode, use the `end-group` command in group configuration mode.

## end-group

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

**Command Default** None

**Command Modes** Group configuration

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** After you have included all configuration statements that you want in a particular configuration group, use the **end-group** command to exit group configuration mode.

Task ID	Task ID	Operation
	config-services	read, write

This example shows how to complete the configuration of a configuration group and exit group configuration mode:

```
RP/0/RP0/CPU0:router(config)# group g-int-gige
RP/0/RP0/CPU0:router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RP0/CPU0:router(config-GRP-if)# mtu 1514
RP/0/RP0/CPU0:router(config-GRP-if)# end-group
RP/0/RP0/CPU0:router(config)#
```

# end-template

To exit template configuration mode and return to XR Config mode, use the **end-template** command in template configuration mode.

## end-template

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

**Command Default** No default behavior or values.

**Command Modes** Template configuration

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the **end-template** command to exit template configuration mode after you have completed the template definition.

To define a template, use the **template** command. To apply a template to the target configuration, use the **apply-template** command. To view the contents of a template, use the **show running-config** command with the optional **template** *template-name* keyword and argument.

Task ID	Task ID	Operations
	config-services	read, write

The following example shows how to enter template configuration mode, define a template named “hostname-template” and then exit from template configuration mode:

```
RP/0/RP0/CPU0:router (config) # template hostname-template
RP/0/RP0/CPU0:router (config-TPL) # hostname router-cs1
RP/0/RP0/CPU0:router (config-TPL) # end-template
RP/0/RP0/CPU0:router (config) #
```

# exclude-group

To exclude (or override) a configuration group (or groups) to be inherited by the router configuration, use the **exclude-group** command in the appropriate configuration mode. To delete the set exclusion, use the **no** form of this command.

**exclude-group** *group-name*

<b>Syntax Description</b>	<i>group-name</i> Configuration group name that needs to be excluded.
---------------------------	-----------------------------------------------------------------------

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	Global configuration
----------------------	----------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Usage Guidelines</b>	More than one configuration group can be excluded simultaneously. A maximum of eight groups can be specified at one time.
-------------------------	---------------------------------------------------------------------------------------------------------------------------



**Note** From Release 6.3.1 onwards, you can enter Flexible CLI config groups, **apply-group** and **exclude-group** command in any order as long as the entire commit has all the group definitions needed.

<b>Task ID</b>	<b>Task ID</b>	<b>Operation</b>
	config-services	read, write

## Example

This example shows how to delete the group G\_interface using the **exclude-group** command:

```
RP/0/RP0/CPU0:router (config) # exclude-group G_interface

exclude-group G_INTERFACE
ipv4 address 12.21.50.100 255.255.0.0
!
interface GigabitEthernet0/0/0/1
ipv4 address 12.21.51.100 255.255.0.0
```

# exit

To close an active terminal session and log off the router, use the **exit** command in XR EXEC mode .

To return the router to the next higher configuration mode, use the **exit** command in any configuration mode.

**exit**

<b>Syntax Description</b>	This command has no keywords or arguments.
---------------------------	--------------------------------------------

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	XR EXEC mode Any configuration
----------------------	-----------------------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Usage Guidelines</b>	<p>To log off from a terminal session, enter the <b>exit</b> command in XR EXEC mode .</p> <p>When exiting from global or administration configuration mode to XR EXEC mode , you are prompted to commit any uncommitted configuration changes.</p>
-------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode .
- If errors are found in the running configuration, the configuration session does not end. To view the errors, enter the **show configuration** (config) command with the **failed** keyword.
- Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.



<b>Note</b>	Entering the <b>exit</b> command from global configuration is functionally equivalent to entering the <b>end</b> command.
-------------	---------------------------------------------------------------------------------------------------------------------------

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	read, write

The following example shows how to return the router to the next higher command mode. In this example, the **exit** command exits from interface configuration mode and returns to global configuration mode. The **exit** command is entered a second time to exit from global configuration mode and return

to XR EXEC mode. Because the configuration has not been committed explicitly (with the **commit** command), the system prompts to commit the configuration changes made during the session.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tengige 0/2/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 1.1.1.1 255.0.0.0
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)# exit
Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]: yes
```

The following example shows how to use the **exit** command from XR EXEC mode to log off from a terminal session:

```
RP/0/RP0/CPU0:router# exit

router con0_RP1_CPU0 is now available

Press RETURN to get started.
```

# hostname

To specify or modify the hostname for the router, use the **hostname** command in XR Config mode.

**hostname** *name*

---

**Syntax Description**     *name* New hostname for the router.

---



---

**Command Default**     The factory-assigned default hostname is “ios.”

---

**Command Modes**     XR Config mode

---

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

---



---

**Usage Guidelines**     The hostname is used in prompts and default configuration filenames.

No blank or space characters are permitted as part of a name. Do not expect case to be preserved. Uppercase and lowercase characters look the same to many Internet software applications. It may seem appropriate to capitalize a name the same way you might do in English, but conventions dictate that computer names appear all lowercase. For more information, see RFC 1178, *Choosing a Name for Your Computer*.

---

Task ID	Task ID	Operations
	root-lr	read, write

---

The following example shows how to change the router hostname:

```
RP/0/RP0/CPU0:router(config)# hostname router1
```



# load

To populate the target configuration with the contents of a previously saved configuration file, use the **load** command in global configuration or administration configuration mode.

**load** *device:directory-path*

<b>Syntax Description</b>	<i>device: directory-path</i> Storage device and directory path of the configuration file to be loaded into the target configuration.				
<b>Command Default</b>	If the full path of the file is not specified, the present working directory is used.				
<b>Command Modes</b>	Global configuration Administration configuration				
<b>Command History</b>	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>Release 7.0.12</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
<b>Usage Guidelines</b>	<p>Use the <b>load</b> command to populate the target configuration with the contents of a previously saved configuration. When loading a file, you must specify the device, directory path, and filename of the configuration file.</p> <p>Use the <b>commit</b> command in conjunction with the <b>load</b> command. Load a new configuration with the <b>load</b> command, and use the <b>commit</b> command with the <b>replace</b> keyword to have the loaded configuration become the active (running) configuration.</p> <p>Use the <b>show configuration failed</b> (config) command with the optional <b>load</b> keyword to display syntax errors that occurred during the last load operation.</p>				
<b>Task ID</b>	<table border="1"> <thead> <tr> <th>Task ID</th> <th>Operations</th> </tr> </thead> <tbody> <tr> <td>config-services</td> <td>read, write</td> </tr> </tbody> </table>	Task ID	Operations	config-services	read, write
Task ID	Operations				
config-services	read, write				

The following example shows how to load a target configuration file into the current configuration session. The current configuration session is then populated with the contents of the file.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# load disk1:myconfig.cfg
RP/0/RP0/CPU0:router(config)# show config

Building configuration...
interface TenGigE 0/3/0/0
  description My 10 GE Interface
  ipv4 address 10.10.11.20 255.0.0.0
!
end
```

# load commit changes

To populate the target configuration with changes from previous configuration commits, use the **load commit changes** command in global configuration or administration configuration mode.

**load commit changes** {*commit-id* | **since** *commit-id* | **last** *number-of-commits*}

Syntax Description		
	<i>commit-id</i>	Specific configuration commit.
	<b>since</b> <i>commit-id</i>	Loads all configuration changes committed into the target buffer since (and including) a specific configuration commit, <i>commit-id</i> .
	<b>last</b> <i>number-of-commits</i>	Loads the configuration changes into the target buffer that have been made during the last number of configuration commits specified with the <i>number-of-commits</i> argument.

**Command Default** None

**Command Modes** Global configuration  
Administration configuration

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the **load commit changes** command to populate the target configuration with changes from previous configuration commits. The changes are not applied until you enter the **commit** command.

Use the **show configuration** (config) command to display the target configuration.

Task ID	Task ID	Operations
	config-services	read, write

The following example shows how to populate the target configuration with changes from a previous configuration commit:

```
RP/0/RP0/CPU0:router(config)# load commit changes since 1000000006
```

```
Building configuration...
Loading.
223 bytes parsed in 1 sec (222)bytes/sec
```

# load configuration failed

To populate the target configuration with the contents of the previous failed configuration commit, use the **load configuration failed** command in global configuration or administration configuration mode.

```
load configuration failed {commit | startup [previous number-of-reloads] [noerror]}
```

Syntax Description	Parameter	Description
	<b>commit</b>	Loads the failed configuration from the last commit.
	<b>startup</b>	Loads the failed configuration from the startup configuration.
	<b>previous <i>number-of-reloads</i></b>	(Optional) Loads the failed configurations from a previous router reload. Valid <i>number-of-reloads</i> values are 1 to 4.
	<b>noerror</b>	(Optional) Excludes the error reasons when the failed configurations are loaded.

**Command Default** None

**Command Modes** Global configuration  
Administration configuration

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the **load configuration failed** command to populate the target configuration with the contents of the previous failed configuration commit.

Task ID	Task ID	Operations
	config-services	read, write

The following example shows how to populate the target configuration with the contents of the previous failed configuration commit:

```
RP/0/RP0/CPU0:router(config)# load configuration failed startup
Loading.
32 bytes parsed in 1 sec (31)bytes/sec
```

# load configuration removed

To populate the target configuration with the contents of the previous removed configuration, use the **load configuration removed** command in global configuration or administration configuration mode.

**load configuration removed** *config-id*

---

## Syntax Description

*config-id* Identifier of the removed configuration to load.

---



---

## Command Default

None

---

## Command Modes

Global configuration

Administration configuration

---

## Command History

### Release

### Modification

Release 7.0.12

This command was introduced.

---



---

## Usage Guidelines

Use the **load configuration removed** command to populate the target configuration with the contents of the removed configuration during installation operations.

---

## Task ID

### Task ID

### Operations

config-services read,  
write

---

The following example shows how to populate the target configuration with the contents of the removed configuration during installation:

```
RP/0/RP0/CPU0:router (config) # load configuration removed 20070316021626.cfg
```

# load rollback changes

To populate the target configuration with the contents of a previous configuration, use the **load rollback changes** command in global configuration or administration configuration mode.

**load rollback changes** {*commit-id* | **last** *number-of-commits* | **to** *commit-id*}

<b>Syntax Description</b>	<i>commit-id</i>	Rolls back the configuration changes for a specific configuration commit.
	<b>last</b> <i>number-of-commits</i>	Rolls back to the configuration that existed before the last number of commits (specified with the <i>number-of-commits</i> argument) were made.
	<b>to</b> <i>commit-id</i>	Rolls back to the running configuration that existed before the configuration specified with the <i>commit-id</i> argument.
<b>Command Default</b>	None	
<b>Command Modes</b>	Global configuration	
	Administration configuration	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	Use the <b>load rollback changes</b> command to load rollback configuration changes to the target configuration. This command is similar to the <b>rollback configuration</b> command. The difference between the commands is that the <b>load rollback changes</b> command copies the rollback changes to the target configuration and does not commit the changes until the changes are explicitly committed with the <b>commit</b> command.	
	Use the <b>show configuration rollback changes</b> command to display rollback changes.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	read, write

The following example shows how to populate the target configuration with the contents of a previous configuration:

```
RP/0/RP0/CPU0:router(config)# load rollback changes 100000004
Building configuration...
Loading.
302 bytes parsed in 1 sec (301)bytes/sec
```

# man

Cisco IOS XR software provides online help for standard command-line interface (CLI) commands using manual (man) pages. To display manual pages, use the **man** command in EXEC mode.

**man** {**command** *command-name* | **feature** [*feature-name*] | **keyword** *keywords*}

## Syntax Description

<b>command</b> <i>command-name</i>	Displays the manual pages for a specific command. The <i>command-name</i> argument must include the complete command name.
<b>feature</b> [ <i>feature-name</i> ]	Displays all commands available in the feature. Use the <b>man</b> command with the <b>feature</b> keyword to list the available feature names.
<b>keyword</b> <i>keywords</i>	Displays a list of command names that match the keywords. Enter one or more keywords to match in a command. When entering multiple keywords, the keywords must be entered in the same sequential order as they are in the command.

## Command Default

None

## Command Modes

EXEC

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

You must have the documentation PIE installed before you can use the **man** command. If you attempt to run this command without the documentation PIE installed, an error is displayed as shown in the following example:

```
RP/0/RP0/CPU0:router# man command show install

Building index table...
Warning. Unable to get directory info for '/pkg/man' :No such file or directory.
Discarding!
man [5521656]:Building index table failed. No entries found
```

For information about installing optional software PIEs, see the *Upgrading and Managing Cisco IOS XR Software* module in *System Management Configuration Guide for Cisco 8000 Series Routers*.

Use the **man** command to display the manual pages for a specific command on the basis of the command name, a feature, or a keyword. Each man page contains the command name, syntax, command mode, usage, examples, and related commands.

The **man** command queries and displays command information about the router. A query can be based on keywords or a feature. The **feature** *feature-name* keyword and argument display all commands that match the feature. For example, entering **man feature -base-1** displays all commands that match the -base-1 feature. The **keyword** *keywords* keyword and argument display all commands that contain the specified keyword. For example, **man keyword ipv4** displays all commands that contain ipv4.

Task ID	Task ID	Operations
	basic-services	read

The following example shows how to display the manual page for the **arp timeout** command:

```
RP/0/RP0/CPU0:router# man command arp timeout
```

```
COMMAND
arp timeout
```

```
DESCRIPTION
```

```
To specify how long dynamic entries learned on an interface remain in the
Address Resolution Protocol (ARP) cache, use the arp timeout command in
interface configuration mode. To remove the arp timeout command from the
configuration file and restore the system to its default condition with
respect to this command, use the no form of this command.
```

```
arp timeout seconds
```

```
no arp timeout<seconds>
```

```
SYNTAX DESCRIPTION
```

```
seconds
```

```
Time, in seconds, for which an entry remains in the ARP cache. The
range is from 0 to 4294967. A value of 0 means that entries are never
cleared from the cache. The default is 14400.
```

```
DEFAULTS
```

```
Entries remain in the ARP cache for 14400 seconds (4 hours).
```

```
COMMAND MODES
```

```
Interface configuration
```

```
COMMAND HISTORY
```

```
Release
Modification
```

```
Release 2.0
This command was introduced.
```

```
USAGE GUIDELINES
```

```
To use the arp timeout command, you must be a member of a user group
```

associated with the cef task ID.

For detailed information about user groups and task IDs, refer to the Configuring AAA Services on Cisco IOS-XR Software module of the Cisco IOS-XR System Security Configuration Guide.

This command is ignored when issued on interfaces that do not use ARP. Also, ARP entries that correspond to the local interface or that are statically configured by the user never time out.

The show interfaces command displays the ARP timeout value in hours:minutes:seconds, as follows:

```
* * * * * START OF LISTING * * * * *
```

```
ARP type: ARPA, ARP Timeout 04:00:00
```

```
* * * * * END OF LISTING * * * * *
```

#### EXAMPLES

The following example shows how to set the ARP timeout to 3600 seconds to allow entries to time out more quickly than the default:

```
* * * * * START OF LISTING * * * * *
```

```
RP/0/RP0/CPU0:router# configure
```

```
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP1/CPU0/0
```

```
RP/0/RP0/CPU0:router(config-if)# arp timeout 3600
```

```
* * * * * END OF LISTING * * * * *
```

#### RELATED COMMANDS

Command

Description

clear arp-cache

Deletes all dynamic entries from the ARP cache.

show arp (cache)

Displays the entries in the ARP table.

show interfaces

Displays statistics for all interfaces configured on the networking device.



# more

To display the contents of a file, use the **more** command in EXEC or administration EXEC mode.

```
more [{/ascii | /binary | /ebcdic}] filesystem:directory-path location [{node-id | all}]{| begin
regular-expression || exclude regular-expression || include regular-expression}
```

Syntax Description	
<b>/ascii</b>	(Optional) Displays a binary file in ASCII format.
<b>/binary</b>	(Optional) Displays a file in hexadecimal or text format.
<b>/ebcdic</b>	(Optional) Displays a binary file in ebcdic format.
<i>filesystem:directory-path</i>	File system location of the file to be displayed. Include the file system alias for the <i>filesystem</i> argument, followed by a colon, and the directory path of the file to be displayed.
<b>location</b> [node-id   all]	(Optional) Displays the contents of a file on a designated node or all nodes.
<i>regular-expression</i>	(Optional) Regular expression found in the file.
	Vertical bar (the “pipe” symbol) indicates that an output processing specification follows.
<b>begin</b>	(Optional) Begins unfiltered output of the <b>more</b> command with the first line that contains the regular expression.
<b>exclude</b>	(Optional) Displays output lines that do not contain the regular expression.
<b>include</b>	(Optional) Displays output lines that contain the regular expression.

**Command Default** None

**Command Modes** EXEC

Administration EXEC

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

Use the **more** command to display any text file, especially an ASCII file stored on the router or accessible through the network. The file can be a configuration file or any other text file.

## Filtering Output

This table shows filter options for the output displayed by the **more** command.

**Table 1: Filtering Options**

Command	Purpose
<b>more</b> <i>filesystem:</i>   <b>begin</b> <i>regular-expression</i>	Begins unfiltered output of the <b>more</b> command with the first line that contains the regular expression.
<b>more</b> <i>filesystem:</i>   <b>exclude</b> <i>regular-expression</i>	Displays output lines that do not contain the regular expression.
<b>more</b> <i>filesystem:</i>   <b>include</b> <i>regular-expression</i>	Displays output lines that contain the regular expression.

## Adding a Filter at the --More-- Prompt

You can also specify a filter at the --More-- prompt of a **more** command output. To filter output from the --More-- prompt, enter a forward slash ( / ) followed by a regular expression. The filter remains active until the command output finishes or is interrupted (using **Ctrl-Z** or **Ctrl-C**).

- A second filter cannot be specified at a --More-- prompt if a filter has already been specified at the original command or at a previous --More-- prompt.
- The minus sign ( - ) preceding a regular expression displays output lines that do not contain the regular expression.
- The plus sign ( + ) preceding a regular expression displays output lines that contain the regular expression.



**Note** After you specify a filter for a **more** command, you cannot specify another filter at the next --More-- prompt. The first specified filter remains until the **more begin** command output finishes or until you interrupt the output. The use of the keyword does not constitute a filter.

## Task ID

### Task ID Operations

filesystem execute

The following example shows partial sample output from the **more** command. The output displays a configuration file saved on the hard disk drive.

```
# more hddisk:/user/alternate.cfg

!! Last configuration change at 15:52:55 UTC Fri Feb 13 2009 by UNKNOWN
!
line console
exec-timeout 0 0
!
interface MgmtEth0/RP1/CPU0/0
  ipv4 address 10.32.45.154 255.0.0.0
!
interface TenGigE0/1/0/0
  ipv4 address 10.32.45.155 255.0.0.0
  keepalive disable
```

```

!
interface TenGigE0/1/0/1
  ipv4 address 10.32.45.156 255.0.0.0
  keepalive disable
!
interface TenGigE0/1/0/2
/ip
  ipv4 address 10.32.45.157 255.0.0.0
  keepalive disable
!
interface TenGigE0/1/0/3
  ipv4 address 10.32.45.158 255.0.0.0
  keepalive disable
!
interface TenGigE0/2/0/0
  ipv4 address 10.32.45.159 255.0.0.0
  keepalive disable
!
--More--

```

The following example shows partial sample output from the **more** command. The output begins with unfiltered output from the first line that contains the regular expression “ipv4.” In this example, a new search is specified that begins with output lines that contain the regular expression “ipv4.”

```

RP/0/RP0/CPU0:router# more disk0:config.backup | begin ipv4

ipv4 address 2.2.2.2 255.255.255.255
!
interface TenGigE0/3/1/0
  shutdown
!
interface TenGigE0/3/1/2
  shutdown
!
interface TenGigE0/2/1/0
  ipv4 address 1.1.1.1 255.255.255.0
  keepalive disable
!
interface TenGigE0/2/1/1
  ipv4 address 1.1.1.1 255.255.255.0
  keepalive disable
!
interface TenGigE0/2/1/2
  ipv4 address 1.1.1.1 255.255.255.0
  keepalive disable
!
interface TenGigE0/2/1/3
  shutdown
!
  /ipv4

filtering...
  ipv4 address 1.1.1.1 255.255.255.0
  proxy-arp disable
  shutdown
!
interface TenGigE 0/1/0/0
  ipv4 address 1.1.1.1 255.255.255.0
  proxy-arp disable
!
route ipv4 0.0.0.0/0 12.25.26.5
route ipv4 223.255.254.254/32 12.25.0.1

```

```
end
```

The following example shows partial sample output of the **more** command on the sample file `config.backup` in `disk0:`. The command usage is `more disk0:config.backup | include log`. At the `--More--` prompt, a new search is specified that begins with output lines that contain the regular expression `“aaa.”`

```
RP/0/RP0/CPU0:router# more disk0:config.backup | include log

logging trap
logging trap informational
logging console debugging
logging history size 1
.
.
.

/aaa

filtering...
aaa authentication login default none
```

The following example shows partial sample output from the **more** command. The output excludes lines that contain the regular expression `“alias.”` In this example, at the `--More--` prompt, a new search is specified, beginning with output lines that contain the regular expression `“ipv4 address.”`

```
RP/0/RP0/CPU0:router# more disk0:myconfig/file | exclude alias

Building configuration...
!! Last configuration change at 18:17:00 UTC Thu May 16 2009 by lab
!
hostname router
line console
  exec-timeout 0 0
  width 132
  length 0
  session-timeout 0
/ipv4 address

filtering...
ipv4 address 10.10.1.1 255.255.255.255
!
interface Loopback200
  ipv4 address 10.20.1.1 255.255.255.255
!
interface TenGigE0/0/0/0
  ipv4 address 10.30.1.1 255.255.0.0
  keepalive 100
!
interface preconfigure TenGigE0/1/0/1
  shutdown
end
```

## pwd (config)

To display the current configuration submode from a configuration submode, use the **pwd** command in any supported configuration submode.

### pwd

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

**Command Default** None

**Command Modes** Any subconfiguration mode

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** No specific guidelines impact the use of this command.

The following example shows how to use the **pwd** command from an interface configuration submode:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tengige 0/6/4/5
RP/0/RP0/CPU0:router(config-if)# pwd

interface TenGigE0/6/4/5
RP/0/RP0/CPU0:router(config-if)#
```

# rollback configuration

To roll back the running configuration to a previous configuration, use the **rollback configuration** command in EXEC or administration EXEC mode.

**rollback configuration** {**last** *number-of-commits* | **to** *commit-id*} {**best-effort** | **force**} [**label** *label*]  
**comment** *comment*

## Syntax Description

<b>last</b> <i>number-of-commits</i>	Rolls back to the configuration that existed before the last number of commits (specified with the <i>number-of-commits</i> argument) were made.
<b>to</b> <i>commit-id</i>	Rolls back to the running configuration that existed before the configuration specified with the <i>commit-id</i> argument.
<b>best-effort</b>	Rolls back to the configuration that existed before the last n commits, and commits only valid changes (best effort). Some configuration changes might fail due to semantic errors.
<b>force</b>	(Optional) Specifies to override any commit blocks.
<b>label</b> <i>label</i>	(Optional) Assigns a text label to this rollback. The <i>label</i> argument must begin with a letter.
<b>comment</b> <i>comment</i>	(Optional) Assigns a text comment to this rollback. The <i>comment</i> argument can be up to 60 characters long.

## Command Default

None

## Command Modes

XR EXEC mode

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

Each time the **commit** command is entered, a commit ID is assigned to the new configuration. You can revert the system to the configuration of a previous commit ID with the **rollback configuration** command:

- Use the **to** keyword to revert to the configuration that existed *before* the configuration specified with the *commit-id* argument.
- Use the **last** keyword to revert to the configuration that existed *before* the last number of configuration commits (specified with the *number-of-commits* argument) were made.
- Use **show configuration commit list** to display a list of the commit IDs available for rollback operations.



**Note** The most recent 100 commits are retained by the system. As new commit IDs are added, the oldest commit IDs are discarded and are no longer available for rollback operations.

Use the **force** keyword to override commits that would fail otherwise. This is useful in the event of a low-memory condition on the router, to revert to a commit that would remove a configuration that caused the low-memory condition.



**Note** The rollback operation may fail if you try to rollback two (or more) commits where the individual commits involve the configuration and removing of the configuration of the same item, and there is a dependency of one item over another in any of the individual commit operations.

Task ID	Task ID	Operations
	root-lr (EXEC)	read, write
	root-system (administration EXEC)	read, write

### Rolling Back to a Specific Commit ID

The following example shows how to roll back to a specific commit ID. In this example, the **show configuration commit list** command displays the available rollback points. The configuration is then rolled back to a prior commit with the **rollback configuration** command.

```
RP/0/RP0/CPU0:router# show configuration commit list

SNo. Label/ID      User      Line      Client      Time Stamp
~~~~ ~~~~~~
1    1000000009 lab      con0_0_C  Rollback    02:41:08 UTC Sun Sep 26 2009
2    1000000008 lab      con0_0_C  CLI         02:40:30 UTC Sun Sep 26 2009
3    1000000007 lab      con0_0_C  CLI         02:39:54 UTC Sun Sep 26 2009
4    1000000006 lab      con0_0_C  Rollback    02:38:40 UTC Sun Sep 26 2009
5    1000000005 lab      con0_0_C  CLI         02:37:35 UTC Sun Sep 26 2009
6    1000000004 lab      con0_0_C  CLI         02:37:04 UTC Sun Sep 26 2009

RP/0/RP0/CPU0:router# rollback configuration to 1000000008

Loading Rollback Changes.
Loaded Rollback Changes in 1 sec
Committing.
1 items committed in 1 sec (0)items/sec
Updating.RP/0/RP0/CPU0:Sep 26 02:42:09.318 : config_rollback[65707]: %LIBTARCFG-
6-COMMIT : Configuration committed by user 'lab'. Use 'show commit changes 100
0000010' to view the changes.
```

```
Updated Commit database in 1 sec  
Configuration successfully rolled back to '1000000008'.
```

### Rolling Back to a Span of Configuration Commits

The following example shows how to roll back to the configuration that existed prior to the last two configuration commits:

```
RP/0/RP0/CPU0:router# rollback configuration last 2
```

```
Loading Rollback Changes.  
Loaded Rollback Changes in 1 sec  
Committing.  
1 items committed in 1 sec (0)items/sec  
Updating.  
Updated Commit database in 1 sec  
Configuration successfully rolled back 2 commits.
```



# root

To return to configuration mode from a configuration submode, use the **root** command in any supported configuration submode.

## root

### Syntax Description

This command has no keywords or arguments.

### Command Default

None

### Command Modes

Any subconfiguration mode except the following:

- The **root** command is not available under the route-policy submodes, because it requires the **end-policy** command to exit out of the configuration.
- The **root** command is not available in template submode, but is available in the submodes configurable under the template submode.

### Command History

Release	Modification
Release 7.0.12	This command was introduced.

### Usage Guidelines

No specific guidelines impact the use of this command.

### Task ID

Task ID	Operations
config-services	read

The following example shows how to use the **root** command to return to configuration mode from the interface configuration submode:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tengige 0/1/0/0
RP/0/RP0/CPU0:router(config-if)# root
RP/0/RP0/CPU0:router(config)#
```

The following example shows how to use the **root** command from a submode configurable under the template submode. In this example, the **root** command is used to return to configuration mode from the username submode:



**Note** The recommended range for a user-defined username is 2-253 characters.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# template test
RP/0/RP0/CPU0:router(config-TPL)# username xyz
RP/0/RP0/CPU0:router(config-un)# root
RP/0/RP0/CPU0:router(config)# show conf
```

```
Building configuration...
template test
  username xyz
  !
end-template
end
```



---

**Tip** The **root** command is not available from the template submode, but is available in the submodes configurable under the template submode.

---

# save configuration

To save the contents of a configuration to a file, use the **save configuration** command in global configuration or administration configuration mode.

**save configuration** [**running**] *device:directory-path*

<b>Syntax Description</b>	<b>running</b> (Optional) Saves the contents of the running configuration.				
	<i>device: directory-path</i> Storage device and directory path of the configuration file to be loaded into the target configuration.				
<b>Command Default</b>	None				
<b>Command Modes</b>	Global configuration Administration configuration				
<b>Command History</b>	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>Release 7.0.12</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
<b>Usage Guidelines</b>	To save a configuration to a file, use the <b>save configuration</b> command. To save a configuration that failed to a file, use the <b>save configuration failed</b> command.				
<b>Task ID</b>	<table border="1"> <thead> <tr> <th>Task ID</th> <th>Operations</th> </tr> </thead> <tbody> <tr> <td>config-services</td> <td>read</td> </tr> </tbody> </table>	Task ID	Operations	config-services	read
Task ID	Operations				
config-services	read				

The following example shows the configuration saved to disk0: from global configuration mode:

```
RP/0/RP0/CPU0:router(config)# save configuration disk0:sample3
Destination file name (control-c to abort): [/sample3]?
Building configuration.
1 lines built in 1 second
[OK]
```

The following example shows the configuration saved to disk1 from administration EXEC mode:

```
RP/0/RP0/CPU0:router(admin-config)# save configuration disk1:sample4
Destination file name (control-c to abort): [/sample4]?
Building configuration.
1 lines built in 1 second
[OK]
```

# save configuration changes

To save the changes of a configuration to a file, use the **save configuration changes** command in global configuration or administration configuration mode.

**save configuration changes** *device:directory-path*

<b>Syntax Description</b>	<i>device: directory-path</i> Storage device and directory path of the configuration file to be loaded into the target configuration.	
<b>Command Default</b>	None	
<b>Command Modes</b>	Global configuration Administration configuration	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	To save the configuration changes to be made during a replace operation to a file, use the <b>save configuration changes</b> command.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	read

The following example shows the configuration saved to disk0: from global configuration mode:

```
RP/0/RP0/CPU0:router(config)# save configuration changes disk0:sample3

Destination file name (control-c to abort): [/sample3]?
Building configuration.
1 lines built in 1 second
[OK]
```

# save configuration commit changes

To save the changes for a commit, or a series of commits, to a file, use the **save configuration commit changes** command in global configuration or administration configuration mode.

```
save configuration commit changes {commit-id | last number-of-commits | since commit-id}
device:directory-path
```

Syntax Description		
	<i>commit-id</i>	Specific commit ID.
	<b>last</b> <i>number-of-commits</i>	Saves changes made in the most recent <i>number-of-commits</i> .
	<b>since</b> <i>commit-id</i>	Saves changes made since (and including) a specific <i>commit-id</i> .
	<i>device: directory-path</i>	Storage device and directory path of the configuration file to be loaded into the target configuration.

**Command Default** None

**Command Modes** Global configuration  
Administration configuration

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the **save configuration commit changes** command to save the changes made in a commit operation to a file. You can specify a specific commit ID, all the changes since a specified commit ID, or the changes that occurred during the last *n* commits.

Task ID	Task ID	Operations
	config-services	read

The following example saves the changes from the last two commit operations to disk0:

```
RP/0/RP0/CPU0:router# save configuration commit changes last 2 disk0:sample1

Destination file name (control-c to abort): [/sample1]?
Building configuration.
5 lines built in 1 second
[OK]
```

# save configuration failed

To save the contents of the failed configuration, use the **save configuration failed** command in global configuration or administration configuration mode.

```
save configuration failed [{load | noerrors | startup [previous number] [noerror]]]
device:directory-path
```

## Syntax Description

<b>load</b>	(Optional) Saves the failed configuration (syntax errors) in the last reload.
<b>noerrors</b>	(Optional) Excludes the error reasons from the saved configuration.
<b>startup</b>	(Optional) Saves the failed configuration during startup.
<b>previous number</b>	(Optional) Saves a failed startup configuration from the specified previous sessions. The <i>number</i> argument is a value between 1 and 4 that indicates how many failed startup configurations to save.
<i>device: directory-path</i>	Storage device and directory path of the configuration file to be saved.

## Command Default

None

## Command Modes

Global configuration  
Administration configuration

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

To save a configuration to a file, use the **save configuration** command.

To save a configuration that failed to a file, use the **save configuration failed** command.

To save a configuration that failed during startup to a file, use the **save configuration failed** command with the **startup** keyword.

## Task ID

Task ID	Operations
config-services	read

The following example saves the failed configuration to disk0:

```
RP/0/RP0/CPU0:router# save configuration failed disk1:/configs
```

# save configuration merge

To save the contents of a merged configuration to a file, use the **save configuration merge** command in global configuration or administration configuration mode.

**save configuration merge** *device:directory-path*

<b>Syntax Description</b>	<i>device : directory-path</i> Storage device and directory path of the configuration file to be loaded into the target configuration.				
<b>Command Default</b>	None				
<b>Command Modes</b>	Global configuration Administration configuration				
<b>Command History</b>	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>Release 7.0.12</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
<b>Usage Guidelines</b>	No specific guidelines impact the use of this command.				
<b>Task ID</b>	<table border="1"> <thead> <tr> <th>Task ID</th> <th>Operations</th> </tr> </thead> <tbody> <tr> <td>config-services</td> <td>read</td> </tr> </tbody> </table>	Task ID	Operations	config-services	read
Task ID	Operations				
config-services	read				

The following example shows the configuration saved to disk0:

```
RP/0/RP0/CPU0:router# save configuration merge disk0:sample3
Destination file name (control-c to abort): [/sample3]?
Building configuration.
1 lines built in 1 second
[OK]
```

# save configuration removed

To save the contents of a removed configuration to a file, use the **save configuration removed** command in global configuration or administration configuration mode.

**save configuration removed** *removed-configuration-file* *device:directory-path*

<b>Syntax Description</b>	<i>removed-configuration-file</i>	Specifies the name of the removed configuration file.
	<i>device:directory-path</i>	Storage device and directory path of the configuration file to be loaded into the target configuration.
<b>Command Default</b>	None	
<b>Command Modes</b>	Global configuration	
	Administration configuration	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	When a package is deactivated, the configuration belonging to that package is removed from the running configuration and saved to a file. To save a copy of the removed configuration file, use the <b>save configuration removed</b> command.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	read

To view a list of the available removed configuration files, use the **save configuration removed** command followed by a question mark:

```
RP/0/RP0/CPU0:router(config)# save configuration removed ?
    20051208042507.cfg  Removed configuration.
    20051208044553.cfg  Removed configuration.
    <cr>
```

In the following example, a removed configuration is saved to disk0: and assigned the filename "sample3:"

```
RP/0/RP0/CPU0:router(config)# save configuration removed 20051208042507.cfg disk0:sample3
Destination file name (control-c to abort): [/sample3]?
Building configuration.
1 lines built in 1 second
[OK]
```



# save rollback changes

To save the rollback changes, use the **save rollback changes** command in global configuration or administration configuration mode.

**save rollback changes** {*commit-id* | **last** *number-of-commits* | **to** *commit-id*} *device:directory-path*

Syntax Description		
	<i>commit-id</i>	Specific commit ID.
	<b>last</b> <i>number-of-commits</i>	Saves the rollback changes for the last <i>n</i> commits
	<b>to</b> <i>commit-id</i>	Saves rollback changes up to a specific <i>commit-id</i> .
	<i>device: directory-path</i>	Storage device and directory path of the configuration file to be loaded into the target configuration.

**Command Default** None

**Command Modes** Global configuration  
Administration configuration

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the **save rollback changes** command to save the changes that would be made in a configuration rollback to a specific commit point or for a series of commits.

Task ID	Task ID	Operations
	config-services	read

The following example shows that the rollback changes for the commit point 5 are saved to the file sample4 on disk0:

```
RP/0/RP0/CPU0:router# save rollback changes last 1 disk0:sample4
```

```
Destination file name (control-c to abort): [/sample4]?
Building configuration.
6 lines built in 1 second
[OK]
```

# service cli commit-optimized enable

To prevent the re-application of the commands which are already present in the running configuration of the router, use the **service cli commit-optimized enable** command in XR Config mode.

**service cli commit-optimized enable**

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

**Command Default** None

**Command Modes** XR Config mode

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** None

## Example

```
Router# configure
Router(config)# service cli commit-optimized enable
Router(config)# commit
```

# set default-afi

To set the default address family identifier (AFI) for the current session, use the **set default-afi** command in EXEC mode.

```
set default-afi {all | ipv4 | ipv6}
```

## Syntax Description

**all** Sets the default AFI to IPv4 and IPv6 for the current session.

**ipv4** Sets the default AFI to IPv4 for the current session. This is the default setting.

**ipv6** Sets the default AFI to IPv6 for the current session.

## Command Default

The default AFI setting is set to IPv4 for all sessions.

## Command Modes

EXEC

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

Use the **set default-afi** command to set the default AFI for the current session. This command acts as a keystroke shortcut for **show** commands. If the default AFI setting is set to IPv4, then you would not have to specify the **ipv4** keyword for **show** commands that support the **ipv4** keyword. For example, if the AFI setting is set to IPv4, you could issue the **show route** command without specifying the **ipv4** keyword to display IPv4 routes in the Routing Information Base (RIB).

Use the **show default-afi-safi-vrf** command to display the default AFI setting.

## Task ID

Task ID	Operations
basic-services	read, write

The following example shows how to set the default AFI to IPv6:

```
RP/0/RP0/CPU0:router# set default-afi ipv6

%% Default Address Family Identifier is set to 'ipv6'
```

## set default-safi

To set the default subaddress family identifier (SAFI) for the current session, use the **set default-safi** command in EXEC mode.

```
set default-safi {all | multicast | unicast}
```

Syntax Description	
<b>all</b>	Sets the default SAFI to multicast and unicast for the current session.
<b>multicast</b>	Sets the default SAFI to multicast for the current session.
<b>unicast</b>	Sets the default SAFI to unicast for the current session. This is the default setting.

Command Default	The default SAFI setting is set to unicast for all sessions.
-----------------	--------------------------------------------------------------

Command Modes	EXEC
---------------	------

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

Usage Guidelines	Use the <b>set default-safi</b> command to set the default SAFI setting for the current session. This command acts as a keystroke shortcut for <b>show</b> commands. If the default SAFI setting is set to unicast, you would not have to specify the <b>unicast</b> keyword for <b>show</b> commands that support that keyword. For example, if the default SAFI setting is set to unicast, you could issue the <b>show router</b> command without specifying the <b>unicast</b> keyword to display information about unicast address prefixes in the Routing Information Base (RIB).
------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Use the **show default-afi-safi-vrf** command to display the default SAFI setting.

Task ID	Task ID	Operations
	basic-services	read, write

The following example shows how to set the default SAFI to multicast:

```
Router# set default-safi multicast

%% Default Sub-Address Family Identifier is set to 'multicast'
```

## set default-vrf

To set the default VPN routing and forwarding (VRF) instance for the current session, use the **set default-vrf** command in EXEC mode.

```
set default-vrf {name | none}
```

### Syntax Description

*name* Default VPN routing and forwarding name.

**none** Sets the default VPN routing and forwarding name to empty.

### Command Default

The default VRF setting is set to empty.

### Command Modes

EXEC

### Command History

Release	Modification
Release 7.0.12	This command was introduced.

### Usage Guidelines

Use the **set default-vrf** command to set the default VRF setting for the current session. This command acts as a keystroke shortcut for **show** commands. For example, if the default VRF is configured, you can issue the **show route** command without specifying the VRF name.

When the default VRF for the session is set to **none**, then IPv4 routes for the system default VRF are displayed.



**Note** To override the default VRF setting, specify the VRF name in the **show** command.

Use the **show default-afi-safi-vrf** command to display the default VRF setting.

In the following example, the default VRF is set to “dft\_vrf:”

```
Router# set default-vrf dft_vrf

%% Default Virtual Routing/Forwarding is set to 'dft_vrf'
```

In the following command, the **show route** command is entered without specifying a VRF name. The results for the "dft\_vrf" VRF are displayed because the default VRF was set to “dft\_vrf.”

```
Router# show route ipv4

% No matching vrf found
```

When the default VRF for the session is set to **none**, the system default VRF routes are displayed. In the following example, the default VRF is set to (empty) and the **show route** command displays the system default VRF information:

```
Router# set default-vrf none
```

```
%% Default Virtual Routing/Forwarding is set to ''
```

```
Router# show route ipv4
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP  
i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2  
ia - IS-IS inter area, su - IS-IS summary null, * - candidate default  
U - per-user static route, o - ODR, L - local
```

```
Gateway of last resort is 12.29.0.1 to network 0.0.0.0
```

```
S* 0.0.0.0/0 [1/0] via 12.29.0.1, 00:31:30  
L 10.10.10.10/32 is directly connected, 3d02h, Loopback1  
C 12.29.0.0/16 is directly connected, 00:31:30, MgmtEth0/0/CPU0/0  
L 12.29.56.21/32 is directly connected, 00:31:30, MgmtEth0/0/CPU0/0
```

# show

To display information about the system configuration or operational state, use the **show** command in EXEC mode, administration EXEC mode, or any configuration mode.

```
show command [{ | begin regular-expression | | exclude regular-expression | | file filesystem: | | include regular-expression }
```

## Syntax Description

<i>command</i>	Supported <b>show</b> command.
	Vertical bar (the “pipe” symbol) indicates that an output processing specification follows.
<i>regular-expression</i>	(Optional) Regular expression found in <b>show</b> command output.
<b>begin</b>	(Optional) Begins unfiltered output of the <b>show</b> command with the first line that contains the regular expression.
<b>exclude</b>	(Optional) Displays output lines that do not contain the regular expression.
<b>file</b> <i>filesystem:</i>	(Optional) Writes the output lines that contain the regular expression to the specified file on the specified file system. Include the file system alias for the <i>filesystem</i> argument, followed by a colon, and the directory path and filename.
<b>include</b>	(Optional) Displays output lines that contain the regular expression.

## Command Default

None

## Command Modes

EXEC

Administration EXEC

Any configuration

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

The **show** commands display information about the system and its configuration. To display a list of the available **show** commands, use the question mark (?) online help function.

### Filtering Output

Search options for the **show** command are shown in this table.

*Table 2: Show Command Search Options*

Command	Purpose
<b>show</b> <i>command</i>   <b>begin</b> <i>regular-expression</i>	Begins unfiltered output of the <b>show</b> command command with the first line that contains the regular expression.

Command	Purpose
<b>show command</b>   <b>exclude</b> <i>regular-expression</i>	Displays output lines that do not contain the regular expression.
<b>show command</b>   <b>include</b> <i>regular-expression</i>	Displays output lines that contain the regular expression.
<b>show command</b>   <b>file filesystem:</b>	Writes the output lines that contain the regular expression to the specified file on the specified file system.

### Adding a Filter at the --More-- Prompt

You can also specify a filter at the --More-- prompt of a **show** command output. To filter output from the --More-- prompt, enter a forward slash (/) followed by a regular expression. The filter remains active until the command output finishes or is interrupted (using **Ctrl-Z** or **Ctrl-C**).

- If a filter is specified at the original command or a previous --More-- prompt, a second filter cannot be applied.
- The use of the **begin** keyword does not constitute a filter.
- The minus sign (-) preceding a regular expression displays output lines that do not contain the regular expression.
- The plus sign (+) preceding a regular expression displays output lines that contain the regular expression.

Task ID	Task ID	Operations
	Task ID for the feature used with the <b>show</b> command	read

For example, the **show interfaces** command requires read privileges in the interface task ID.

The following example shows output from the **show interface | include protocol** command. In this example, the **show** command includes only lines in which the regular expression “protocol” appears:

```
RP/0/RP0/CPU0:router# show interface | include protocol

Null0 is up, line protocol is up
0 drops for unrecognized upper-level protocol
TenGigE0/2/0/0 is administratively down, line protocol is administratively down
0 drops for unrecognized upper-level protocol
TenGigE0/2/0/1 is administratively down, line protocol is administratively down
0 drops for unrecognized upper-level protocol
TenGigE0/2/0/2 is administratively down, line protocol is administratively down
0 drops for unrecognized upper-level protocol
TenGigE0/2/0/3 is administratively down, line protocol is administratively down
0 drops for unrecognized upper-level protocol
FastEthernet0/RP0/CPU0/0 is administratively down, line protocol is administratively
down
FastEthernet0/RP0/CPU0/0 is administratively down, line protocol is administratively
down
0 drops for unrecognized upper-level protocol
```

On most systems, the **Ctrl-Z** key combination can be entered at any time to interrupt the output and return to EXEC mode. For example, use the **show running-config | begin hostname** command to



start the display of the running configuration file at the line containing the hostname setting, then use **Ctrl-Z** when you get to the end of the information you are interested in.

The following example shows sample output from the **show configuration running | begin line** command. The output begins with unfiltered output from the first line that contains the regular expression “line.” In this example, at the **--More--** prompt, a new search is specified that begins with output lines that contain the regular expression “ipv4.”



---

**Note** The use of the **begin** keyword does not constitute a filter.

---

```
RP/0/RP0/CPU0:router# show configuration running | begin line

Building configuration...
line console
  exec-timeout 120 120
!
logging trap
--More--
/ipv4

filtering...
route ipv4 0.0.0.0 255.255.0.0 pos0/2/0/0
interface TenGigE0/2/0/0
  ipv4 address 172.19.73.215 255.255.0.0
end
```

# show aliases

To display all defined aliases or the aliases defined in a specified mode, use the **show aliases** command in EXEC mode.

**show aliases**

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

**Command Default** Displays all aliases currently configured on the system.

**Command Modes** EXEC

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the **show aliases** command to display all aliases currently configured on the system.

Task ID	Task ID	Operations
	basic-services	read

The following example illustrates sample output from the **show aliases** command. The output displays a summary of all the command aliases configured.

```
RP/0/RP0/CPU0:router# show aliases

exec mode aliases:
ipv4_brief          show ipv4 interface brief

interface mode aliases:
sample_int         tengige 0/2/0/0
```

# show apply-group

To display the applied configuration groups, use the **show apply-group** command in EXEC mode.

## show apply group

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

**Command Default** None

**Command Modes** EXEC

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** The output of this show command indicates if the group is a part of the global apply-group or is a part of the sub-mode level apply-group.

Task ID	Task ID	Operation
	config-services	read

## Example

This example shows how to use the **show apply-group** command:

```
RP/0/RP0/CPU0:router # show apply-group
                        Global      Non-Global
Groups Reference Count Reference Count
-----
B                1                0
C                1                0
```

Reference count can either be 0 or 1. 0 indicates that the group is not applied globally; 1 indicates that the group is globally applied.

# show configuration failed

To display information about a configuration that failed during the last commit, use the **show configuration failed** command in EXEC mode.

**show configuration failed** [**inheritance**]

<b>Syntax Description</b>	<b>inheritance</b> Displays the failed configuration details at the inheritance level.
---------------------------	----------------------------------------------------------------------------------------

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	EXEC
----------------------	------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Usage Guidelines</b>	Without the inheritance keyword, this command displays the failed configuration information in brief.
-------------------------	-------------------------------------------------------------------------------------------------------

<b>Task ID</b>	<b>Task ID</b>	<b>Operation</b>
	config-services	read

## Example

This example shows how to run the **show configuration failed** command:



<b>Note</b>	When there are two (or more) groups that have failed, the ordering of the failed groups is displayed in the same order as the apply-group statement.
-------------	------------------------------------------------------------------------------------------------------------------------------------------------------

```
RP/0/RP0/CPU0:router (config) # show config failed
!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.
apply-group GROUP-1 GROUP-2 GROUP-3 GROUP-4 GROUP-5
!% Please issue "show configuration failed inheritance" for details.
Applying following groups failed: GROUP-2 GROUP-4 GROUP-5
```

# show configuration changes

To display the configuration changes to be made during a replace operation, use the **show configuration changes** command in global configuration or administration configuration System Admin Config mode .

**show configuration changes** [**diff**]

<b>Syntax Description</b>	<b>diff</b> (Optional) Displays the changes in UNIX-like format.						
<b>Command Default</b>	None						
<b>Command Modes</b>	XR Config mode System Admin Config mode						
<b>Command History</b>	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>Release 7.0.12</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	Release 7.0.12	This command was introduced.		
Release	Modification						
Release 7.0.12	This command was introduced.						
<b>Usage Guidelines</b>	No specific guidelines impact the use of this command.						
<b>Task ID</b>	<table border="1"> <thead> <tr> <th>Task ID</th> <th>Operations</th> </tr> </thead> <tbody> <tr> <td>config-services</td> <td>read</td> </tr> <tr> <td>basic-services</td> <td>read</td> </tr> </tbody> </table>	Task ID	Operations	config-services	read	basic-services	read
Task ID	Operations						
config-services	read						
basic-services	read						

The following example shows the changes to be made during a replace operation:

```
RP/0/RP0/CPU0:router(config)# show configuration changes diff
```

```
Building configuration...
# hostname router
# hostname bla
- logging console
- telnet vrf default ipv4 server disable
- domain ipv4 host xhu-u5
- domain ipv4 host coax-u10
- domain ipv4 host coax-u10.cisco.com
- domain name
- interface Loopback1
- ipv4 address 10.0.0.2 255.255.255.224
- !
- interface Loopback2
- description
- !
- interface Loopback5
- description
- !
- interface Loopback6
- description
- !
- interface MgmtEth0/0/CPU0/0
```

```
- ipv4 address 10.0.0.1 255.255.255.224
- !
- interface GigabitEthernet0/2/0/0
- shutdown
- !
- interface GigabitEthernet0/2/0/1
- shutdown
- !
- interface GigabitEthernet0/2/0/2
- shutdown
- !
- router static
- address-family ipv4 unicast
-   0.0.0.0/0 255.255.255.224
- !
- !
end
```

# show configuration commit changes

To display the changes made to the running configuration by previous configuration commits, a configuration commit, or for a range of configuration commits, use the **show configuration commit changes** command in EXEC, administration EXEC, administration configuration, or global configuration mode.

**show configuration commit changes** {*commit-id* | **since** *commit-id* | **last** *number-of-commits* | **all** } [**diff**]

Syntax Description		
	<b>since</b>	Displays all changes committed to the running configuration since (and including) a specific configuration commit.
	<i>commit-id</i>	Displays configuration changes for a specific configuration commit.
	<b>last</b> <i>number-of-commits</i>	Displays the changes made to the running configuration during the last number of configuration commits specified for the <i>number-of-commits</i> argument.
	<b>all</b>	Displays commit ID and configurations completed for last 100 commits.
	<b>diff</b>	(Optional) Displays added lines, changed lines, and deleted lines.

**Command Default** None

**Command Modes** EXEC  
Administration EXEC  
Administration configuration  
Global configuration

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Each time a configuration is committed with the **commit** command, the configuration commit operation is assigned a commit ID. The **show configuration commit changes** command displays the configuration changes made since the specified commit.

To display a list of the available commit IDs, enter the **show configuration commit list** command. You can also display the commit IDs by entering the **show configuration commit changes** command with the online help function (?).

You can't view commit IDs from a different release if the syntax or semantics of the configuration changed in the current release.



**Note** Syntax of a configuration refers to its structure and format, while the semantics of a configuration refers to its backend interpretation.

Task ID	Task ID	Operations
	config-services	read

The following example shows sample output from the **show configuration commit changes** command. The output displays commit IDs.

```
RP/0/RP0/CPU0:router# show configuration commit list

SNo. Label/ID   User      Line      Client      Time Stamp
~~~~ ~~~~~~    ~~~~     ~~~~~     ~~~~~~     ~~~~~~
1    1000000077   lab      con0_1_C  CLI         15:42:45 UTC Fri Jan 30 2009
2    1000000076   lab      con0_1_C  Rollback    15:30:39 UTC Fri Jan 30 2009
3    1000000075   lab      con0_1_C  Rollback    15:25:26 UTC Fri Jan 30 2009
4    1000000074   lab      con0_1_C  Rollback    15:04:29 UTC Fri Jan 30 2009
5    1000000073   lab      con0_1_C  CLI         14:49:07 UTC Fri Jan 30 2009
6    1000000072   lab      con0_1_C  CLI         14:48:35 UTC Fri Jan 30 2009
```

The following example shows sample output from the **show configuration commit changes** command with the *commit-id* argument. In this example, the output displays the changes made in the configuration commit assigned commit ID 1000000077.

```
RP/0/RP0/CPU0:router# show configuration commit changes 1000000077

Building configuration...
alias exec shrun show configuration running
alias exec shver show version
end
```

The following example shows sample output from the **show configuration commit changes** command with the **since** *commit-id* keyword and argument. In this example, the output displays the configuration changes made since the configuration commit assigned commit ID 1000000077 was committed.

```
RP/0/RP0/CPU0:router# show configuration commit changes since 1000000077

Building configuration...
no hw-module node 0/RP0/CPU0 shutdown
hostname router
logging trap
no logging console
logging history size 1
alias exec shrun show configuration running
alias exec shver show version
interface MgmtEth0/RP1/CPU0/0
  ipv4 address 12.25.34.10 255.255.0.0
```



```

no shutdown
!
interface preconfigure MgmtEth0/RP0/CPU0/0
no shutdown
!
no route ipv4 0.0.0.0/0 12.7.0.1
route ipv4 0.0.0.0/0 12.25.0.1
route ipv4 223.255.254.254/32 12.25.0.1
telnet ipv4 server enable
end

```

The following example shows sample output from the **show configuration commit changes** command with the **diff** keyword. In the display, the following symbols signify changes:

+ indicates an added line.

– indicates a deleted line.

# indicates a modified line.

```
RP/0/RP0/CPU0:router# show configuration commit changes last 1 diff
```

```

Building configuration...
+ interface Loopback1000
+ ipv4 address 190.190.180.1 255.255.255.255
!
end

+ interface Loopback1000
+ ipv4 address 190.190.180.1 255.255.255.255
!
end

```

The following example shows sample output from the **show configuration commit changes** command with the **all** keyword. In this example, the output displays the list of configurations that are committed in last 100 commits along with their commit-ID.

```
RP/0/RP0/CPU0:router# show configuration commit changes all
Commit ID : 1000000001
```

```

-----
Building configuration...
!! IOS XR Configuration 0.0.0
interface GigabitEthernet0/0/0/1
speed 100
!
end

```

```
Commit ID : 1000000002
```

```

-----
Building configuration...
!! IOS XR Configuration 0.0.0
interface GigabitEthernet0/0/0/1
no speed 100
!
end

```

```
Commit ID : 1000000003
```

```

-----
Building configuration...

```

**show configuration commit changes**

```
!! IOS XR Configuration 0.0.0
interface GigabitEthernet0/0/0/0
!
shutdown
end
```

# show configuration commit list

To display information about the configuration commits stored in the commit database, use the **show configuration commit list** command in EXEC, administration EXEC, administration configuration, or global configuration mode.

**show configuration commit list** [*number-of-commits*] [**detail**]

<b>Syntax Description</b>	<i>number-of-commits</i> (Optional) Number of commits (beginning with the most recent commit) that are available for rollback.
	<b>detail</b> (Optional) Displays detailed commit information, including comments.

**Command Default** If this command is entered without any optional arguments or keywords, the output displays information about all the configuration commits stored in the commit database.

**Command Modes**

- EXEC
- Administration EXEC
- Administration configuration
- Global configuration

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the **show configuration commit list** command to list the commit IDs (up to 100) that are available for rollback.



**Note** The most recent 100 commits are retained by the system. As new commit IDs are added, the oldest commit IDs are discarded and are no longer available for rollback operations.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	read

The following example shows sample output from the **show configuration commit list** command. The output displays the commit IDs that are available for rollback.

```
RP/0/RP0/CPU0:router# show configuration commit list

SNo. Label/ID      User      Line      Client      Time Stamp
~~~~~ ~~~~~~
1    1000000010 UNKNOWN  con0_0_C  Rollback    02:25:53 UTC Fri Feb 06 2009
2    1000000009 UNKNOWN  con0_0_C  CLI         02:23:09 UTC Fri Feb 06 2009
```

## show configuration commit list

```

3      1000000008 UNKNOWN con0_0_C CLI      02:22:54 UTC Fri Feb 06 2009
4      1000000007 UNKNOWN con0_0_C CLI      02:22:18 UTC Fri Feb 06 2009
5      1000000006 UNKNOWN con0_0_C CLI      02:07:21 UTC Fri Feb 06 2009

```

The following table describes the significant fields shown in the display.

**Table 3: show configuration commit list Field Descriptions**

Field	Description
SNo.	Serial number of the commit entry.
Label/ID	If a label was assigned to a commit, the first 10 characters of the label display; otherwise, the autogenerated commit ID displays.
User	User who executed the commit.
Line	Line in which the user session was established. In some cases, this field may display "UNKNOWN" or "SYSTEM". These fields indicate that an internal commit was made by the system.
Client	The management interface used to make the commit.
Time Stamp	Time and date when the commit was executed.

# show configuration (config)

To display information about the current configuration session (target configuration), use the **show configuration** command in any configuration mode.

**show configuration** [**merge**] [**running**]

<b>Syntax Description</b>	<b>merge</b> (Optional) Displays the configuration that occurs if the contents of the uncommitted changed (target configuration) are committed to the running configuration.				
	<b>running</b> (Optional) Displays the running (committed) configuration.				
<b>Command Default</b>	When the <b>show configuration</b> command is entered without an argument, the uncommitted changes to the target configuration are displayed.				
<b>Command Modes</b>	Any configuration				
<b>Command History</b>	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>Release 7.0.12</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
<b>Usage Guidelines</b>	<p>Use the <b>show configuration</b> command to display details on uncommitted configuration changes.</p> <p>Use the <b>show configuration</b> command with the <b>running</b> keyword to display the running (active) configuration.</p> <p>Prior to committing the target configuration, use the <b>show configuration</b> command with the <b>merge</b> keyword from any configuration mode to display the result of merging the target configuration with the running configuration.</p>				
<b>Task ID</b>	<table border="1"> <thead> <tr> <th>Task ID</th> <th>Operations</th> </tr> </thead> <tbody> <tr> <td>basic-services</td> <td>read</td> </tr> </tbody> </table>	Task ID	Operations	basic-services	read
Task ID	Operations				
basic-services	read				

In this example, the **show configuration** command displays uncommitted changes made during a configuration session:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tengige0/3/0/3
RP/0/RP0/CPU0:router(config-if)# description faq
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.10.11.20 255.0.0.0
RP/0/RP0/CPU0:router(config-if)# show configuration

Building configuration...
interface TenGigE0/3/0/3
description faq
ipv4 address 10.10.11.20 255.0.0.0
end
```

The following example shows sample output from the **show configuration** command with the optional **merge** keyword. The command is entered during a configuration session. The output displays the result of merging the target and running configuration, without committing the changes.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tengige0/3/0/3
RP/0/RP0/CPU0:router(config-if)# description faq
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.10.11.20 255.0.0.0
RP/0/RP0/CPU0:router(config-if)# show configuration merge

Building configuration...
hostname router
interface TenGigE0/0/0/0
  ipv4 address 1.2.3.4 255.0.0.0
  exit
interface TenGigE0/3/0/3
  description faq
  ipv4 address 1.1.1.1 255.0.0.0
  shutdown
end
```

# show configuration failed incompatible

To display any configurations that were removed from the running configuration because they were not understood by the software being activated, use the **show configuration failed incompatible** command in EXEC or administration EXEC mode.

**show configuration failed incompatible**

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

**Command Default** None

**Command Modes** EXEC  
Administration EXEC

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Any configurations in the running configuration that are not understood by new software being installed are removed from the running configuration. To see which configurations were removed, use the **show configuration failed incompatible** command.

Task ID	Task ID	Operations
	config-services	read

# show configuration failed remove

To display information about a configuration that failed while being removed during installation operations, use the **show configuration failed remove** command in EXEC or administration EXEC mode.

**show configuration failed remove**

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

**Command Default** None

**Command Modes** EXEC  
Administration EXEC

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** No specific guidelines impact the use of this command.

Task ID	Task ID	Operations
	config-services	read

The following example shows a failed commit operation:

```
RP/0/RP0/CPU0:router# show configuration failed remove

!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.

multicast-routing
 no address-family ipv4
  !!% Process did not respond to sysmgr
 address-family ipv4
  no interface all enable
 !!% Process did not respond to sysmgr
 !
 !
```

Because the configuration failed to be removed, it is still displayed in the output from the **show running-configuration** command as expected:

```
RP/0/RP0/CPU0:router# show running-configuration
...
router pim vrf default address-family ipv4
 auto-rp candidate-rp GigabitEthernet0/2/0/3 scope 255 group-list 224/4 interval 10
 !
 multicast-routing
```



```
address-family ipv4
  interface all enable
!
```

# show configuration failed rollback

To display information about a configuration that failed in the last rollback operation, use the **show configuration failed rollback** command in EXEC or administration EXEC mode.

**show configuration failed rollback**

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

**Command Default** None

**Command Modes** EXEC  
Administration EXEC

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** No specific guidelines impact the use of this command.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	read
	root-lr	read

# show configuration failed startup

To display information about a configuration that failed at startup, use the **show configuration failed** command in EXEC or administration EXEC mode.

**show configuration failed startup** [{noerror | previous *number*}]

<b>Syntax Description</b>	<p><b>noerror</b> (Optional) Displays the configuration that failed at startup without an error reason.</p> <p><b>previous <i>number</i></b> (Optional) Displays the previous failed startup configuration or configurations. The <i>number</i> argument is a value from 1 to 4, which displays the failed startup configurations in previous of sessions.</p>				
<b>Command Default</b>	If no keywords are specified, this command displays the details of the failed startup configuration including error reasons.				
<b>Command Modes</b>	EXEC Administration EXEC				
<b>Command History</b>	<table border="1"> <thead> <tr> <th data-bbox="386 934 803 970">Release</th> <th data-bbox="803 934 1528 970">Modification</th> </tr> </thead> <tbody> <tr> <td data-bbox="386 991 803 1026">Release 7.0.12</td> <td data-bbox="803 991 1528 1026">This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
<b>Usage Guidelines</b>	No specific guidelines impact the use of this command.				
<b>Task ID</b>	<table border="1"> <thead> <tr> <th data-bbox="386 1138 544 1173">Task ID</th> <th data-bbox="544 1138 1528 1173">Operations</th> </tr> </thead> <tbody> <tr> <td data-bbox="386 1194 544 1230">config-services</td> <td data-bbox="544 1194 1528 1230">read</td> </tr> </tbody> </table>	Task ID	Operations	config-services	read
Task ID	Operations				
config-services	read				

# show configuration history

To display a history of configuration events, use the **show configuration history** command in EXEC, administration EXEC, administration configuration, or global configuration mode.

```
show configuration history [{alarm | backup | cfs-check | commit | rebase | shutdown | startup}]
[{first number | last number | reverse}] [detail]
```

## Syntax Description

<b>alarm</b>	(Optional) Displays alarm events.
<b>backup</b>	(Optional) Displays configuration backup events.
<b>cfs-check</b>	(Optional) Displays CFS check events.
<b>commit</b>	(Optional) Displays commit events.
<b>rebase</b>	(Optional) Displays commit database consolidation events.
<b>shutdown</b>	(Optional) Displays shutdown events.
<b>startup</b>	(Optional) Displays startup events, including alternate configurations, failed configurations, and other events.
<b>first number</b>	(Optional) Displays the first <i>x</i> number of events, where <i>x</i> is the <i>number</i> argument.
<b>last number</b>	(Optional) Displays the last <i>x number</i> events. Replace with the number of events to display.
<b>reverse</b>	(Optional) Displays the most recent events first.
<b>detail</b>	(Optional) Displays detailed information, including comments.

## Command Default

When entered without any optional arguments or keywords, this command displays all configuration events. The oldest events are displayed at the top of the list for each event type.

## Command Modes

EXEC  
Administration EXEC  
Administration configuration  
Global Configuration

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

Use the **show configuration history** command to display information about the last (up to) 1500 configuration events.

Use one of the available keywords to display the configuration event only for that event type. Use the **first number** and **last number** keywords and arguments to display a specified number of events. Use the **reverse** keyword to display the newest events at the top of the list.

Task ID	Task ID	Operations
	config-services	read

In the following example, the **show configuration history** command is used to display the history of all configuration events for an SDR:

```
RP/0/RP0/CPU0:router# show configuration history
```

Sno.	Event	Info	Time Stamp
~~~~	~~~~~	~~~~	~~~~~
1	alarm	inconsistency alarm raised	Thu Jun 22 15:23:15 2009
2	startup	configuration applied	Thu Jun 22 15:23:32 2009
3	OIR config	restore	Thu Jun 22 15:23:25 2009
4	OIR config	restore	Thu Jun 22 15:23:33 2009
5	OIR config	restore	Thu Jun 22 15:23:33 2009
6	OIR config	restore	Thu Jun 22 15:23:34 2009
7	OIR config	restore	Thu Jun 22 15:23:34 2009
8	OIR config	restore	Thu Jun 22 15:23:35 2009
9	OIR config	restore	Thu Jun 22 15:23:36 2009
10	OIR config	restore	Thu Jun 22 15:23:37 2009
11	OIR config	restore	Thu Jun 22 15:23:37 2009
12	OIR config	restore	Thu Jun 22 15:23:38 2009
13	OIR config	restore	Thu Jun 22 15:23:38 2009
14	OIR config	restore	Thu Jun 22 15:23:39 2009
15	OIR config	restore	Thu Jun 22 15:23:39 2009
16	OIR config	restore	Thu Jun 22 15:23:40 2009
17	OIR config	restore	Thu Jun 22 15:23:40 2009
18	OIR config	restore	Thu Jun 22 15:23:42 2009
19	OIR config	restore	Thu Jun 22 15:23:42 2009
20	OIR config	restore	Thu Jun 22 15:23:42 2009
21	OIR config	restore	Thu Jun 22 15:23:43 2009

--More--

In the following example, the **show configuration history** command is used to display only the startup configuration events:

```
RP/0/RP0/CPU0:router# show configuration history startup
```

Sno.	Event	Info	Time Stamp
~~~~	~~~~~	~~~~	~~~~~
1	startup	configuration applied	Thu Jun 22 15:23:32 2009
2	startup	configuration applied	Sat Jul 1 15:02:24 2009
3	startup	configuration applied	Sat Jul 8 17:36:52 2009
4	startup	configuration applied	Sun Jul 9 13:40:27 2009
5	startup	configuration applied	Sat Jul 15 18:18:54 2009

In the following example, the **show configuration history** command with the **commit detail** keywords is used to display additional details regarding the commit events:

```
RP/0/RP0/CPU0:router# show configuration history commit detail
```

```
1) Event: commit          Time: Thu Jun 22 15:44:33 2009
   Commit ID: 1000000001 Label:
   User: lab              Line: vty0
   Client: CLI            Comment:

2) Event: commit          Time: Thu Jun 22 16:58:18 2009
```

```

Commit ID: 1000000002 Label:
User: lab           Line: vty2
Client: CLI         Comment:

3) Event: commit           Time: Thu Jun 22 16:58:39 2009
   Commit ID: 1000000003 Label:
   User: lab           Line: vty2
   Client: CLI         Comment:

4) Event: commit           Time: Sat Jul  1 15:29:31 2009
   Commit ID: 1000000001 Label:
   User: lab           Line: vty0
   Client: CLI         Comment:

5) Event: commit           Time: Sat Jul  1 15:32:25 2009
   Commit ID: 1000000002 Label:
   User: lab           Line: vty0
--More--

```

**Table 4: show configuration history Field Descriptions**

Field	Description
SNo.	Serial number of the entry.
Event	Type of configuration event.
Info	Summary of the configuration action.
Time Stamp	Time and date when the event was run.
Label/ID	If a label was assigned to a commit, the first 10 characters display; otherwise, the autogenerated commit ID displays.
User	User who issued the command.
Line	Line in which the user session was established. In some cases, this field may display “UNKNOWN” or “SYSTEM”. These fields indicate that an internal action was made by the system.
Client	The management interface used to make the event.

# show configuration inconsistency replica

To display any configuration inconsistencies on a replica node, use the **show configuration inconsistency replica** command in EXEC or administration EXEC mode.

**show configuration inconsistency replica location *node-id* [detail]**

<b>Syntax Description</b>	<b>location <i>node-id</i></b> Displays any configuration inconsistencies on the designated node. The <i>node-id</i> argument is expressed in the <i>rack/slot</i> notation.
	<b>detail</b> Displays a detailed list of inconsistencies.

**Command Default** Administration EXEC mode: Displays configuration inconsistencies for the admin plane configuration.  
EXEC mode: Displays configuration inconsistencies for an SDR configuration.

**Command Modes** EXEC  
Administration EXEC

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** In administration EXEC or EXEC mode, the replica node for the **show configuration inconsistency replica** command is the standby designated system controller (DSC).

Use the **show configuration inconsistency replica** command, before performing a manual switchover or DSC migration, to verify that the node in line to take over for the DSC is in good shape. If any problems are reported, use the **clear configuration inconsistency replica** command to correct them.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	read

The following example shows a configuration with inconsistencies:

```
RP/0/RSP0/CPU0:router# show configuration inconsistency replica location 0/rsp1/cpu0
The replica at location 0/RSP1/CPU0 is inconsistent.
Please run 'clear configuration inconsistency replica location 0/RP1/CPU0'.
```

The following example shows sample output after the inconsistencies have been resolved:

```
RP/0/RSP0/CPU0:Router# show configuration inconsistency replica location 0/rsp1/cpu0
Replica is consistent
```

# show configuration persistent

To display the persistent configuration, use the **show configuration persistent** command in EXEC mode.

**show configuration persistent [diff]**

## Syntax Description

**diff** (Optional) Displays the difference between the running configuration and persistent configuration. This option is available only on the DSC.

## Command Default

If no argument is specified, the **show configuration persistent** command displays the entire contents of the persistent configuration file.

## Command Modes

EXEC

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

The persistent configuration is the configuration stored in nonvolatile memory, from which the running configuration is restored after the router is reloaded. The running configuration should be the same as the persistent configuration. Use the **show configuration persistent** command with the **diff** keyword to check if there is a difference between the running configuration and the persistent configuration.

## Task ID

Task ID	Operations
config-services	read

The following example shows that there is no difference between the running configuration and the persistent configuration:

```
RP/0/RP0/CPU0:router# show configuration persistent diff

Building configuration...
end
```

The following example shows a difference between the running configuration and the persistent configuration:

```
RP/0/RP0/CPU0:router# show configuration persistent diff

Building configuration...
router vrrp
interface tengige0/1/0/1.1
vrrp 1 preempt delay 300
!
interface tengiget0/1/0/1.2
vrrp 1 preempt delay 300
!
interface tengige0/1/0/1.3
```



```
vrrp 1 preempt delay 300
```

# show configuration removed

To display a configuration removed during installation operations, use the **show configuration removed** command in EXEC or administration EXEC mode.

**show configuration removed** *config-id*

<b>Syntax Description</b>	<i>config-id</i> Name of removed configuration. Type (?) to see a list of the names of all removed configurations.	
<b>Command Default</b>	None	
<b>Command Modes</b>	EXEC Administration EXEC	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	No specific guidelines impact the use of this command.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	read

The following example shows a removed configuration:

```
RP/0/RP0/CPU0:router# show configuration removed 20060301112919.cfg

xml agent corba
http server
end
```

# show configuration rollback changes

To display changes that would be made by the **rollback configuration** command or to display the list of commit IDs, use the **show configuration rollback changes** command in EXEC, administration EXEC, administration configuration, or global configuration mode.

**show configuration rollback changes** {*commit-id* | **to** *commit-id* | **last** *number-of-commits*} [**diff**]

Syntax Description		
	<i>commit-id</i>	Name of configuration. When a specific <i>commit-id</i> is specified, only the changes that would occur if only the specified commit is rolled back are displayed.
	<b>to</b> <i>commit-id</i>	Displays the changes that will occur to the running configuration if the system is rolled back to the configuration specified with the <i>commit-id</i> argument.
	<b>last</b> <i>number-of-commits</i>	Displays the changes that will occur to the running configuration if the system is rolled back to the last number of commits specified with the <i>number-of-commits</i> argument.
	<b>diff</b>	(Optional) Displays added lines, changed lines, and deleted lines.

**Command Default** None

**Command Modes** EXEC  
Administration EXEC  
Administration configuration  
Global configuration

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

## Usage Guidelines



**Note** The most recent 100 commits are retained by the system. As new commit IDs are added, the oldest commit IDs are discarded and are no longer available for rollback operations.

Use the *commit-id* argument without the **to** keyword to display the changes for a particular commit. This can be useful for troubleshooting actions of the **rollback configuration** command.

Task ID	Task ID	Operations
	config-services	read

The following example shows sample output from the **show configuration rollback changes** command with the **to** *commit-id* keyword and argument. The output displays the configuration

changes that would occur if the configuration were to be rolled back to the configuration commit specified for the argument.

```
RP/0/RP0/CPU0:router# show configuration rollback changes to 1000000007

Building configuration...
hostname old-name
end
```

The following example shows sample output from the **show configuration rollback changes** command **last** *number-of-commits* keyword and argument. The output displays the configuration changes that would occur if the configuration were to be rolled back to the number of configuration commits specified for the argument.

```
RP/0/RP0/CPU0:router# show configuration rollback changes last 2

Building configuration...
hostname orig_name
interface POS0/1/0/1
  shutdown
!
end
```

The following example shows sample output from the **show configuration rollback changes** command with the **diff** keyword.

In the display, the following symbols signify changes:

- + indicates an added line.
- - indicates a deleted line.
- # indicates a modified line.

```
RP/0/RP0/CPU0:router
show configuration rollback changes last 1 diff

Building configuration...
  interface Loopback1000
#  ipv4 address 1.1.1.1 255.255.255.255
!
end
```

# show configuration running

To display the running configuration, use the **show configuration running** command in the appropriate mode.

```
show configuration running [config-keyword]
```

## Syntax Description

*config-keyword* (Optional) Specific configuration to display.

## Command Default

None

## Command Modes

Administration EXEC  
Administration configuration  
Global configuration

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

Use the **show configuration running** command to display the currently active configuration.

## Task ID

Task ID	Operations
basic-services	read

This example shows the currently running (committed) configuration from administration mode.

```
RP/0/RP0/CPU0:router(admin)# show configuration running

Building configuration...
username lab
  secret 5 $1$XNwt$j8RscNdncKSRoMSnqSpbj/
  group root-system
!
end
```

# show configuration running-config

To display the running configuration, use the **show configuration running-config** command in EXEC mode.

**show configuration running-config** [*config-keyword*]

<b>Syntax Description</b>	<b>inheritance</b>	(Optional) Displays the configuration inherited from any applied configuration group.
	<b>no-annotation</b>	(Optional) Suppresses the display of inheritance messages, when the <b>inheritance</b> keyword is used.
	<i>config-keyword</i>	(Optional) Specific configuration to display.
<b>Command Default</b>	None	
<b>Command Modes</b>	EXEC	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the *config-keyword* argument to display the running configuration for a specific keyword only.

## Display the Configuration from Configuration Groups

By default, if configuration groups are applied in the configuration, they are displayed as they are configured. For example:

```
RP/0/RP0/CPU0:router# show configuration running-config
!
group G-INTERFACE-MTU
  interface 'GigabitEthernet.*'
    mtu 1500
end-group
!
interface GigabitEthernet1/0/0/7
  apply-group G-INTERFACE-MTU
!
```

To display the actual configuration as inherited from any applied configuration groups, use the **inheritance** keyword:

```
RP/0/RP0/CPU0:router# show configuration running-config inheritance
!
interface GigabitEthernet1/0/0/7
  ## Inherited from group G-INTERFACE-MTU
  mtu 1500
!
```

Use the **no-annotation** keyword to suppress the display of the Inheritance messages, "## Inherited from group ...".

Task ID	Task ID	Operations
	basic-services	read

This example shows the currently running (committed) configuration:

```
RP/0/RP0/CPU0:router# show configuration running-config

Building configuration...

    !! Last configuration change at 15:36:31 UTC Thu Nov 17 2009 by lab
    sessions Users with active configuration sess
!n
hostname router
line consolestartup Sh
  exec-timeout 0 0configuration
!
logging console debugging
  |      Ou
snmp-server community public RW
  <cr>
RP/0/0/
ipv4 source-routeadmin)#show confi
key chain IPSLA ?
key 10
  key-string password 1
  ipv4 address 10.0.0.0 255.255.255.0
  encapsulation ppp
  keepalive disable
!
interface POS0/7/0/0
  shutdown
!
interface POS0/7/0/1
  shutdown
!
interface POS0/7/0/2
  shutdown
!
interface POS0/7/0/3
  shutdown
!
route ipv4 0.0.0.0/0 12.7.0.1
ipsla
  responder
!
!
end
```

# show configuration sessions

To display the active configuration sessions, use the **show configuration sessions** command in EXEC or administration EXEC mode.

**show configuration sessions** [**detail**]

<b>Syntax Description</b>	<b>detail</b> (Optional) Displays detailed information.
---------------------------	---------------------------------------------------------

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	EXEC Administration EXEC
----------------------	-----------------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Usage Guidelines</b>	Use the <b>show configuration sessions</b> command to display the active configuration sessions. Use the <b>clear configuration sessions</b> command to clear a configuration session. The <b>show configuration sessions</b> command can be used with the <b>clear configuration sessions</b> command to verify that an active configuration session was cleared.
-------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	read

The following example shows sample output from the **show configuration sessions** command:

```
RP/0/RP0/CPU0:router# show configuration sessions
Current Configuration Session Line      User      Date                               Lock
00000050-001200bb-00000000    con0_5_CPU cisco     Fri Feb 16 17:23:47 2007
```

**Table 5: show configuration sessions Field Descriptions**

Field	Description
Session	System-generated configuration session ID number.
Line	Line in which the user session was established. In some cases, this field may display “UNKNOWN” or “SYSTEM.” These fields indicate that an internal commit was made by the system.
User	User who initiated the configuration session.
Date	Time and date the configuration session was started.



Field	Description
Lock	Locked running-configuration. An asterisk (*) displayed in this field means the session has been locked. Only one session can lock the running configuration at a time.

## show default-afi-safi-vrf

To display the default address family identifier (AFI), subaddress family identifier (SAFI), and VPN routing and forwarding (VRF) instance for the current session, use the **show default-afi-safi-vrf** command in EXEC mode.

**show default-afi-safi-vrf**

<b>Syntax Description</b>	This command has no keywords or arguments.
---------------------------	--------------------------------------------

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	EXEC
----------------------	------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Usage Guidelines</b>	Use the <b>show default-afi-safi-vrf</b> command to display the default AFI and SAFI settings for the current session. The AFI and SAFI settings are controlled by the following commands:
-------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- [set default-afi, on page 59](#)
- [set default-safi, on page 60](#)
- [set default-vrf, on page 61](#)

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	basic-services	read

The following example shows sample output from the **show default-afi-safi-vrf** command:

```
RP/0/RP0/CPU0:router# show default-afi-safi-vrf

%% Default AFI/SAFI/VRF for this session is:
  Address Family Identifier:      'ipv4'
  Sub-Address Family Identifier:  'unicast'
  Virtual Routing/Forwarding:    ''
```

# show history

To display a history of commands executed in EXEC, administration EXEC, administration configuration, or global configuration mode use the **show history** command in one of the supported modes.

**show history** [**detail**]

<b>Syntax Description</b>	<b>detail</b> (Optional) Displays detailed history information.						
<b>Command Default</b>	None						
<b>Command Modes</b>	EXEC Administration EXEC Administration configuration Global configuration						
<b>Command History</b>	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>Release 7.0.12</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	Release 7.0.12	This command was introduced.		
Release	Modification						
Release 7.0.12	This command was introduced.						
<b>Usage Guidelines</b>	The <b>show history</b> command displays a history of the command entered for the current command mode. For example, enter the <b>show history</b> command to display a history of commands entered in EXEC mode. Enter the <b>show history</b> command in global configuration mode to display a history of the commands entered in global configuration mode.						
<b>Task ID</b>	<table border="1"> <thead> <tr> <th>Task ID</th> <th>Operations</th> </tr> </thead> <tbody> <tr> <td>config-services</td> <td>read</td> </tr> <tr> <td>basic-services</td> <td>read</td> </tr> </tbody> </table>	Task ID	Operations	config-services	read	basic-services	read
Task ID	Operations						
config-services	read						
basic-services	read						

In the following example, the **show history** command is run in EXEC mode to display a history of the command entered in EXEC mode:

```
RP/0/RP0/CPU0:router# show history
configure
admin
show history
```

In the following example, the **show history** command is run in global configuration mode to display a history of the command entered in global configuration mode:

```
RP/0/RP0/CPU0:router(config)# show history
interface pos 0/1/0/0
ipv4 address 10.0.0.0
root
end
```

```
describe line default autocommand config
line default autocommand configure
end
show history
```

# template

To create a template name and enter template configuration mode, use the **template** command in global configuration mode. To remove a template definition, use the **no** form of this command.

**template** *name*  
**no template** *name*

<b>Syntax Description</b>	<i>name</i> Unique name for the template to be created.
---------------------------	---------------------------------------------------------

<b>Command Default</b>	No templates are defined.
------------------------	---------------------------

<b>Command Modes</b>	Global configuration
----------------------	----------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines**

Use the **template** command to enter template configuration mode. From template configuration mode, you can group a subset of configuration commands in a named template. Commonly used sets of configuration commands can be grouped into a named template. Defining a template is similar to creating a C macro function. A template provides modularity and ease of use during user configuration.

Use the **end-template** command to exit template configuration mode. After defining a template, use the **apply-template** command to apply the template. Use the **show running-config** command with the optional **template** keyword and *template-name* argument to display the contents of a template.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	config-services	read, write

The following example shows how to enter template configuration mode to create a template. In this example, a template named “pre-pos” is defined for the preconfigured Packet-over-SONET/SDH (POS) interface 0/1/0/1. The **end-template** command is used to exit from template configuration mode.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# template pre-pos
RP/0/RP0/CPU0:router(config-TPL)# interface preconfigure pos0/1/0/0
RP/0/RP0/CPU0:router(config-if-pre)# ipv4 address 10.3.32.154 255.0.0.0
RP/0/RP0/CPU0:router(config-if-pre)# end-template
RP/0/RP0/CPU0:router(config)#
```



---

**Note** After configuring a template, you may want to display the contents of the configured template. To display a template configuration, use the **show running-config** command with the **template name** keyword and argument.

---

The following example shows sample output from the **show running-config** command with the **template name** keyword and argument. In this example, the output displays the contents of a template named “pre-pos.”

```
RP/0/RP0/CPU0:router# show running-config template pre-pos

template pre-pos
  interface preconfigure POS0/1/0/0
    ipv4 address 10.3.32.154 255.0.0.0
  !
end-template
```