



Configuration and File System Management

This module describes methods for configuration management and file transfer enhancements.

- [Auto-Save Configuration, on page 1](#)
- [Increasing Commit Limit, on page 2](#)
- [Fault Recovery Handling, on page 5](#)

Auto-Save Configuration

You can configure the router to automatically take the backup of the running configuration by using **configuration commit auto-save** command. This auto-save feature saves the configuration to the specified location on the router after every **commit** is made. These auto-save files are stored in the form of Linux files.

Configure Auto-Save

Use the **configuration commit auto-save** command to auto save the configuration.

```
Router#configure  
Router(config)#configuration commit auto-save  
Router(config-cfg-autosave)#commit
```

Increasing Commit Limit

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Increasing Commit Limit	Release 7.3.6	<p>The maximum number of commits is increased in the router that allows you to configure complex topology changes without interruptions caused by the default blocking of commit changes during rebase or ASCII backup operations. You can prevent the commit operation from getting blocked by using the cfs check command, which increases the commit (pacount) count from 20 to 40, and the commit file diff size (configuration data) from 2 MB to 4 MB, and by using the clear configuration ascii inconsistency command, which performs an ASCII backup after 55 minutes.</p> <p>The feature modifies the following commands:</p> <ul style="list-style-type: none"> • cfs check • clear configuration ascii inconsistency

The Cisco IOS XR Routers use a two-stage configuration model. The first stage is target configuration, where you build the configurations using necessary commands in the command line interface. The second stage is the commit, where the configuration made in the target stage is added to the router configuration using the **commit** command. After each commit, the router generates a file for the newly configured changes and adds it to its running configuration, making it an integral part of the running configuration.



Note This target configuration does not impact the router's running configuration.

The Cisco IOS XR routers perform rebase and ASCII backup operations to maintain the real time configuration in the backup copy. The rebase and ASCII backup operations block you from committing configurations to the router.

In rebase, the router automatically saves your changes to the backup configuration file after 20 commits or 2 MB of configuration data. The router blocks the commit while saving the configuration to the backup file. The router takes up to 40 seconds to complete rebase operation, during which, if you terminate the CLI session, the router loses the target configurations in the blocked commit.

In ASCII backup, the router automatically saves a copy of its running configuration in the ASCII format. This backup process takes place if there has been a commit to the router configuration, and when the ASCII backup timer completes a 55-minute window after the previous backup event. However, if there was no commit when the ASCII backup timer completes 55 minutes, the counter is reset without any backup. During the ASCII backup, the router blocks the configuration commits.

Starting with Release 7.3.6, we have made the following enhancements:

- You can use the **cfs check** command to increase the rebase limits in the router from 20 to 40 commits and the configuration data from 2 MB to 4 MB. When configuring the router, you can check the current commits count and configuration data size using the **show cfmgr commitdb** command. If the commits count is 20 or higher, or the configuration data size is 2 MB or above, the router will initiate a rebase within 10 seconds. By using the **cfs check** command to increase the commit count to 40 and the configuration data to 4 MB, you can commit without delay.
- You can use the **clear configuration ascii inconsistency** command to perform an ASCII backup and reset the ASCII backup timer to zero. Once the backup is complete, the router will automatically initiate the operation only after 55 minutes since the last commit was performed after the ASCII backup.

Guidelines and Restrictions for Increasing the Commit Limit

- The **cfs check** command increases the rebase limits only for one instance. After executing the **cfs check** command, the router will perform a rebase operation after 40 commits or when the configuration data reaches 4 MB. Once the router performs a rebase operation, the limits will reset to the default values of 20 commits and 2 MB configuration data. To enable 40 commits and 4 MB configuration data, you must perform the **cfs check** command again.
- After executing the **cfs check** command, if a router switches over to standby RP, the rebase limits are retained as 40 commits and configuration data of 4 MB. However, if the router reloads, the rebase limits are reset to 20 commits and 2 MB of configuration data. For example, after executing the **cfs check** command, if the router switches over to standby RP after 30 commits, it will still have ten more commits before a rebase. However, if the router reloads, the rebase limits are reset to default 20 commits and 2 MB of configuration data.
- The **clear configuration ascii inconsistency** command initiates an ASCII backup and resets the ASCII backup timer count to zero. Following this, the router will automatically begin the ASCII backup operation only 55 minutes after the last commit performed after the ASCII backup. For example, if you execute a commit operation after executing a **clear configuration ascii inconsistency** command, the router will perform an ASCII backup operation 55 minutes after the first commit operation. Hence, before the next 55 minutes, you must execute the **clear configuration ascii inconsistency** command again to reset the ASCII backup timer to zero.
- When the router enters standby mode or reloads, the ASCII timer does not reset to zero, and the router performs an ASCII backup operation 55 minutes after the first commit operation before the standby mode or reload.

Increasing the Rebase Limits

You can increase the rebase limits as follows:

1. Use the **cfs check** command to increase the commit count to 40 and the configuration data to 4 MB.

Fault Recovery Handling

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Fault Recovery Handling	Release 7.3.6	<p>You can now configure the number of fault recovery attempts by a line card, fabric card or a route processor before it permanently shuts down, thus preventing a faulty card from entering into a cycle of automatic recovery.</p> <p>This feature introduces the following change:</p> <p>CLI:</p> <ul style="list-style-type: none"> • hw-module fault-recovery <p>YANG DATA Model:</p> <ul style="list-style-type: none"> • New XPath for Cisco-IOS-XR-hw-module-cfg.yang (see Github, YANG Data Models Navigator)

In the previous releases, if a line card, fabric card or a route processor experienced a fault, they used to trigger fault recovery and reboot themselves to be operational. Fault recovery mechanism was time based as the fault recovery count used to reset to zero if the card remained operational for more than hour. After the fault recovery count exceeded five, then the faulty card was shut down. As power related faults triggered were not frequent, and fault recovery count used to reset to zero, the card never entered the shut down mode. As a result the card always attempted for fault recovery.

With the Cisco IOS XR Software Release 7.3.6, we have introduced the **hw-module fault-recovery** command with which you can set the number of times a fault recovery can take place before permanently shutting down a faulty card.



Note This configuration is not applicable for BMC instance

How to Configure the Fault Recovery Attempts

Configuration Examples

The configuration example shows how to configure the fault recovery attempts on the fabric card FC0.

```
Router#configure
Router (config)#hw-module fault-recovery location 0/FC0 count 1
Router (config)#commit
```

Verification

Use **show running-config formal | include hw-module** command to display the number of times a card can initiate recovery attempts before shutting down .

```
Router#show running-config formal | include hw-module
Building configuration...
hw-module fault-recovery location 0/FC0 count 1
```

The following system log is generated when the number of fault recovery attempts on the card exceeds the configured count:

```
RP/0/RP0/CPU0:Dec 4 15:44:25.247 PST: shelfmgr[121]: %PLATFORM-SHELFMGR-4-CARD_SHUTDOWN :
Shutting down 0/FC0: Fault retry attempts exceeded configured count(1)
```

Use the **show reboot history** command to get the reason of card shutting down. In the following example, it shows that the card was shut down due to **Fault retry attempts exceeded configured count(1)**.

```
RP/0/RP0/CPU0:ios#show reboot history location 0/FC0 detail
Mon Dec 4 15:44:55.827 PST
-----
No   Attribute      Value
-----
1    Time (PST)     Dec 04 2023 15:44:22
     Cause Code    0x0800000d
     Cause String  REBOOT_CAUSE_FM
     Graceful Reload No
     Kdump Requested No
     Reason       Fault retry attempts exceeded configured count(1)
```

Use the **show platform** command to see the current state of the card that was shut down because of Fault recovery handling feature.

```
RP/0/RP0/CPU0:ios#show platform
Mon Oct 2 21:08:03.383 UTC

Node           Type                State                Config state
-----
0/RP0/CPU0     8800-RP (Active)    IOS XR RUN          NSHUT
0/RP0/BMC0     8800-RP             OPERATIONAL         NSHUT
0/RP1/CPU0     8800-RP (Standby)  IOS XR RUN          NSHUT
0/RP1/BMC0     8800-RP             OPERATIONAL         NSHUT
0/3/CPU0       8800-LC-48H        IOS XR RUN          NSHUT
0/FC0         8812-FC            SHUT DOWN          NSHUT
0/FC3         8812-FC             OPERATIONAL         NSHUT
0/FT0         SF-D-12-FAN        OPERATIONAL         NSHUT
0/FT1         SF-D-12-FAN        OPERATIONAL         NSHUT
0/FT2         SF-D-12-FAN        OPERATIONAL         NSHUT
0/FT3         SF-D-12-FAN        OPERATIONAL         NSHUT
0/PT0         FAM7000-ACHV-TRAY  OPERATIONAL         NSHUT
0/PT1         FAM7000-ACHV-TRAY  OPERATIONAL         NSHUT
0/PT2         FAM7000-ACHV-TRAY  OPERATIONAL         NSHUT
Router#
```