

Boot Commands

- reload, on page 2
- show epm trace boot, on page 4
- show reboot, on page 6

reload

To reloads the route processor (RP), use the **reload** command in XR EXEC mode.

reload

Syntax Description

This command has no keywords or arguments.

Command Default

None

Command Modes

XR EXEC mode

Command History

Releases	Modifications
Release 7.0.12	This command was introduced.

Usage Guidelines

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

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

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

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

```
Router# reload

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

Task ID

Task ID	Operations
root-lr	execute

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

```
Router# reload
```

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

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

rommon B1 >
```

show epm trace boot

To display execution path monitoring traces, use the **show epm trace boot** command in administration EXEC mode.

show epm trace boot [**hexdump**] [**last** *n*] [**reverse**] [**stats**] [**tailf**] [**unique**][**verbose**] [**wrapping**][**file** *filename* **original**] [**location** {*node-id* | **all**}]

Syntax Description

hexdump	(Optional) Displays traces in hexadecimal format.
last *n*	(Optional) Displays the last *n* number of traces only.
reverse	(Optional) Displays the most recent traces first.
stats	(Optional) Displays execution path statistics.
tailf	(Optional) Displays new traces as they are added.
unique	(Optional) Displays unique entries only, along with the count of the number of times this entry appears.
verbose	(Optional) Displays additional internal debugging information.
wrapping	(Optional) Displays wrapping entries.
file *filename* **original**	(Optional) Specifies the filename of the file to display. You can specify up to four trace files.
location {*node-id* | **all**}	(Optional) Specifies the node of the . The *node-id* argument is entered in the *rack/slot* notation. You can specify up to four nodes. The **all** keyword specifies all nodes.

Command Default

None

Command Modes

Administration EXEC

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

The **show epm trace boot** command provides a simple way of tracking and time-stamping critical events to clearly understand their temporal relationship to one another and the amount of time spent performing critical operations.

Task ID

Task ID	Operations
basic services	read

The following example shows sample output from the **show epm trace boot** command:

```
RP/0/RP0/CPU0:router(admin)# show epm trace boot

Mon Jun  1 03:16:36.946 PST
22 wrapping entries (1024 possible, 0 filtered, 22 total)
Oct  8 07:54:49.610 epm/boot 0/RP0/CPU0 t1  @ 00:00:06 - [init] process-start
Oct  8 07:55:25.710 epm/boot 0/RP0/CPU0 t1  @ 00:00:42 - [insthelper] process-start
Oct  8 07:57:08.992 epm/boot 0/RP0/CPU0 t1  @ 00:02:25 - [sysmgr] process-start
Oct  8 07:57:09.785 epm/boot 0/RP0/CPU0 t7  @ 00:02:26 - [sysmgr] start-level: start
Oct  8 07:57:10.722 epm/boot 0/RP0/CPU0 t1  @ 00:02:27 - [sw_dwnld_svr] process-start
Oct  8 07:57:12.482 epm/boot 0/RP0/CPU0 t11 @ 00:02:29 - [sysmgr] start-level: admin
Oct  8 07:57:13.385 epm/boot 0/RP0/CPU0 t1  @ 00:02:30 - [instdir] process-start
Oct  8 07:57:19.638 epm/boot 0/RP0/CPU0 t1  @ 00:02:36 - [instdir_lr] process-start
Oct  8 07:58:07.045 epm/boot 0/RP0/CPU0 t9  @ 00:03:23 - [sysmgr] admin-plane-up
Oct  8 07:58:52.057 epm/boot 0/RP0/CPU0 t4  @ 00:04:08 - [cfgmgr-rp] admin-config-start
Oct  8 07:58:59.973 epm/boot 0/RP0/CPU0 t4  @ 00:04:16 - [cfgmgr-rp] admin-config-done
Oct  8 07:59:00.079 epm/boot 0/RP0/CPU0 t9  @ 00:04:16 - [sysmgr] start-level: infra
Oct  8 07:59:00.615 epm/boot 0/RP0/CPU0 t1  @ 00:04:17 - [devc-conaux] exec-available
Oct  8 07:59:02.288 epm/boot 0/RP0/CPU0 t4  @ 00:04:18 - [cfgmgr-rp] admin-plane-mount-done
Oct  8 07:59:08.157 epm/boot 0/RP0/CPU0 t6  @ 00:04:24 - [instdir] ready-for-requests
Oct  8 07:59:15.999 epm/boot 0/RP0/CPU0 t6  @ 00:04:32 - [sysmgr] start-level: active
Oct  8 07:59:32.300 epm/boot 0/RP0/CPU0 t13 @ 00:04:48 - [sysmgr] start-level: final
Oct  8 07:59:38.143 epm/boot 0/RP0/CPU0 t9  @ 00:04:54 - [sysmgr] lr-plane-up
Oct  8 07:59:38.189 epm/boot 0/RP0/CPU0 t4  @ 00:04:54 - [cfgmgr-rp] lr-config-start
Oct  8 07:59:49.898 epm/boot 0/RP0/CPU0 t4  @ 00:05:06 - [cfgmgr-rp] lr-config-done
Oct  8 07:59:50.259 epm/boot 0/RP0/CPU0 t4  @ 00:05:06 - [cfgmgr-rp]
bulk-interface-config-start
Oct  8 07:59:50.351 epm/boot 0/RP0/CPU0 t7  @ 00:05:06 - [cfgmgr-rp] node-config-done
```

In this sample output, the time stamp following the @ sign is the elapsed time in the format hh:mm:ss since the execution phase started (for example, since node start, in the case of a boot).

show reboot

To display reboot information for a node, use the **show reboot** command in

EXEC or administration EXEC

System Admin EXEC

mode.

show reboot {**history** [**reverse**] | {**first** | **last**} {**crashinfo** | **syslog** | **trace**} | **graceful**} **location** *node-id*

Syntax Description

first	(Optional) Displays information about the first ungraceful reboot.
last	(Optional) Displays information about the last ungraceful reboot.
crashinfo	Displays crash information for an ungraceful reboot.
syslog	Displays the syslogs related to an ungraceful reboot.
trace	Displays trace information for an ungraceful reboot.
graceful	Displays information about the last graceful reboot.
history	Displays the reboot history of a specific node.
reverse	(Optional) Displays the reboot history information in reverse chronological order. **Note** Starting from Cisco IOS XR Release 24.3.1, the **reverse** keyword is deprecated and will not be supported in future releases.
location *node-id*	Specifies which node to reload. The *node-id* argument is expressed in the *rack/slot* notation.

Command Default

None

Command Modes

EXEC

Administration EXEC

System Admin EXEC

Command History

Release	Modification
Release 24.3.1	The **reverse** keyword is deprecated and will not be supported in future releases. Hence the **show reboot history reverse location** command is also not supported.
Release 7.0.12	This command was introduced.

Usage Guidelines

- The **show reboot history** command displays all reboot causes stored for previous node resets.
- The **show reboot history** command output does not include information about the system reload event triggered by power cycle.
- Crash information (**crashinfo**), syslog, and kernel dumper ltrace (**trace**) is displayed for the first or last reboot if it is an ungraceful reboot.

Task ID

Task ID	Operations
system	read

This example shows the history of reloaded nodes.

```
RP/0/RP0/CPU0:ios#show reboot history location 0/RP0/CPU0
Wed Apr 17 16:55:20.748 PDT
--------------------------------------------------------------------------------
No DATE TIME (PDT) Cause Code Cause String
--------------------------------------------------------------------------------
1 Apr 12 2024 12:01:04 0x00000024 REBOOT_CAUSE_UPGRADE
2 Mar 29 2024 00:08:40 0x00000024 REBOOT_CAUSE_UPGRADE
3 Mar 05 2024 03:58:00 0x00000025 REBOOT_CAUSE_ADMIN
4 Feb 27 2024 22:58:08 0x00000024 REBOOT_CAUSE_UPGRADE
5 Feb 27 2024 00:02:34 0x00000024 REBOOT_CAUSE_UPGRADE
6 Feb 15 2024 11:06:58 0x00000024 REBOOT_CAUSE_UPGRADE
7 Feb 06 2024 18:15:40 0x00000025 REBOOT_CAUSE_ADMIN
8 Feb 02 2024 16:57:24 0x00000024 REBOOT_CAUSE_UPGRADE
--------------------------------------------------------------------------------
```

This example displays the crash information for the first reboot.

```
RP/0/RP0/CPU0:router# show reboot first crashinfo location 0/RP0/CPU0

Crashinfo Timestamp: Thu Jul 19 20:32:57 2007

20070719 20:32:57

Crash Reason: Cause code 0x21000010 Cause: Missed deadline,
  client: sc-reddrv-main, timeout: 5 Process: wd-critical-mon
Traceback: fc1941a0 fc194290 48200738 482013cc 48201c04 fc1d4fb0 Timezone UTC0

Exception at 0xfc1944c8 signal 5 c=1 f=3

Active process(s):
pkg/bin/wd-critical-mon Thread ID 1 on cpu 0
pkg/bin/l3test Thread ID 0 on cpu 1

       REGISTER INFO
      r0        r1        r2        r3
R0   01000000  4817e8c0  4820e208  000000de
       r4        r5        r6        r7
R4   fc1b4856  7fffffff  4817e738  fc1b4856
       r8        r9       r10       r11
R8   00000000  602cf522  00000000  00000000
      r12       r13       r14       r15
R12  602cf51c  4820e1a0  00000000  00000000
      r16       r17       r18       r19
R16  00000000  00000000  00000000  00000000
      r20       r21       r22       r23
```

```
R20  00000000  00000000  48200000  48200000
       r24       r25       r26       r27
R24  48200000  48200000  48200000  48200000
       r28       r29       r30       r31
R28  00000028  00000001  21000010  6029b000
         cnt        lr       msr        pc
R32  00000000  fc194290  0002d932  fc1944c8
         cnd       xer
R36  44000094  20000006

                SUPERVISOR REGISTERS

               Memory Management Registers

               Instruction BAT Registers
                Index #                Value
               IBAT0U #              0x1ffe
               IBAT0L #                0x12
               IBAT1U #                   0
               IBAT1L #                   0
               IBAT2U #          0x30000ffe
               IBAT2L #          0xf0000032
               IBAT3U #          0xfffc0003
               IBAT3L #             0x40011

               Data BAT Registers
                Index #                Value
               DBAT0U #              0x1ffe
               DBAT0L #                0x12
               DBAT1U #                   0
               DBAT1L #          0x10000012
               DBAT2U #          0x30000ffe
               DBAT2L #          0xf000006a
               DBAT3U #          0xfffc0003
               DBAT3L #             0x40011

               Segment Registers
                Index #              SR-Value
                    0 #                    0
                    1 #                    0
                    2 #                    0
                    3 #                    0
                    4 #                    0
                    5 #                    0
                    6 #                    0
                    7 #                    0
                    8 #                    0
                    9 #                    0
                   10 #                    0
                   11 #                    0
                   12 #                    0
                   13 #                    0
                   14 #                    0
                   15 #                    0

                Exception Handling Registers
          Data Addr Reg #                DSISR
            0x602cf440 #          0x42000000
        SPRG0 #       SPRG1 #       SPRG2 #       SPRG3
        0x1 # 0x21000010 # 0x6029b000 #        0
     SaveNRestore SRR0 #    SaveNRestore SRR1
          0xfc1944c4 #             0x2d932

  Miscellaneous Registers
```

```
    Processor Id Reg #                  0
                HID0 #          0x8410c0bc
                HID1 #          0x9001ac80

               MSSCR0 #             0x88000
               MSSSR0 #                   0

 STACK TRACE
#0 0xfc194290
#1 0x48200738
#2 0x482013cc
#3 0x48201c04
#4 0xfc1d4fb0
```