

# Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Troubleshoot](#)

[Linecard booted as expected](#)

[Fabric card unable to boot due to driver error](#)

[Card stuck in POWEROFF or PRESENT STATE](#)

[Card stuck in POWER\\_ON STATE](#)

[Card stuck in SW\\_INACTIVE STATE](#)

[Scenario 1: SW\\_EVENT\\_FAILURE: SW\\_EVENT\\_ADMIN\\_VM\\_FAILURE reported by shelf\\_mgr syslogs](#)

[Scenario 2: LC stuck in FAILED state, last event: HW\\_EVENT\\_FAILURE fail\\_code=LC\\_POWER\\_MAIN\\_FAULT](#)

[Card stuck in UNKOWN STATE](#)

[Using RCONSOLE tool](#)

[List of commands to be collected before opening TAC case](#)

[Related Cisco Support Community Discussions](#)

## Introduction

This document describes how to troubleshoot Network Convergence System 6000 (NCS6K) Line card boot failures. In addition, it also provides an overview of data that can be collected which could help TAC in detailed investigation.

## Prerequisites

### Requirements

Cisco recommends that you have basic knowledge of XR Command Line Interface (CLI).

### Components Used

This document was created using XR release versions 5.0.1, 5.2.1, 5.2.3 and 5.2.4.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

## Background Information

If Route Processor (RP), Fabric Card (FC) or Line Card (LC) is not booting and getting stuck in boot stage, first step should be to start collecting 'show tech ctrace' from Sysadmin Virtual Machine (VM). This show tech provides information on the current system state and interaction between different components. However, there is a chance that Sysadmin also known as Calvados is not up on the card and 'show tech ctrace' will not be able to collect information for the affected card. This will occur because card is not reachable via SSH. For such cases, rconsole procedure will be necessary to know why card is stuck during its boot process.

**Note:** This file is usually pretty big (500MB-1GB) and will be stored on the Sysadmin VM. In order to be extract it from the box, it must be copied to the XR VM (instructions provided later in the same document).

## Troubleshoot

Check status of each card on the Sysadmin Virtual Machine (VM) and find out its current state. Pay special attention to H/w and S/w State. Note that cards that shows S/w state as N/A are CPU less cards (FC cards, Fan Tray, etc.) which are essentially controlled by RP CPU. Cards in "OPERATIONAL" state are CPU based cards and therefore have software loaded.

```
sysadmin-vm:0_RP0# show platform
```

Location	Card Type	HW State	SW State	Config State
0/0	PROTO-CXP-1XPITA	OPERATIONAL	OPERATIONAL	NSHUT
<b>0/2</b>	<b>PROTO-CXP-2XPITA</b>	<b>POWERED_OFF</b>	<b>SW_INACTIVE</b>	<b>NSHUT</b>
0/3	NC6-10X100G-M-K	OPERATIONAL	OPERATIONAL	NSHUT
0/RP0	NC6-RP	OPERATIONAL	OPERATIONAL	NSHUT
<b>0/RP1</b>	<b>NC6-RP</b>	<b>POWERED_ON</b>	<b>SW_INACTIVE</b>	<b>NSHUT</b>
0/FC0	NC6-FC	POWERED_ON	N/A	NSHUT
0/FC1	NC6-FC-MC	POWERED_ON	N/A	NSHUT
<b>0/FC2</b>	<b>UNKNOWN</b>	<b>FAILED</b>	<b>N/A</b>	<b>NSHUT</b>
0/FC3	NC6-FC	POWERED_ON	N/A	NSHUT
0/FC4	NC6-FC-B2B	POWERED_ON	N/A	NSHUT
0/FC5	NC6-FC	OPERATIONAL	N/A	NSHUT
0/FT0	NC6-FANTRAY	OPERATIONAL	N/A	NSHUT
0/FT1	NC6-FANTRAY	OPERATIONAL	N/A	NSHUT
0/PT0	NCS-AC-PWRTRAY	OPERATIONAL	N/A	NSHUT
0/PT1	NCS-AC-PWRTRAY	OPERATIONAL	N/A	NSHUT
0/PT2	NCS-AC-PWRTRAY	OPERATIONAL	N/A	NSHUT

Next step is to check Card Chip Controller (CCC) inventory command and confirm the card state.

```
sysadmin-vm:0_RP0# show controller ccc inventory summary
```

CCC Inventory Summary :

Location	Card Type	BP ID	Serial Number	HW Ver	Card State
<b>0/RP0</b>	<b>NC6-RP (master)</b>	<b>0</b>	<b>SAD15270129</b>	<b>0.1</b>	<b>CARD_READY</b>
0/RP1	NC6-RP (slave)	1	SAD1527012P	0.1	CARD_READY
0/FC0	NC6-FC	8	SAD1618002F	0.2	WAIT_DEV_INIT
0/FC1	NC6-FC	9	SAD153901ZT	0.2	WAIT_DEV_INIT
0/FC4	NC6-FC	12	SAL1803KQEY	1.0	PON_POWERING_UP
0/FC5	NC6-FC	13	SAD16180043	0.2	WAIT_DEV_INIT
0/0	NC6-10X100G-M-K	16	SAL1650UCN9	0.4	PXE_BOOTING
0/4	NC6-10X100G-M-K	20	SAD154502XU	0.1	CARD_READY

Following are the different scenarios which can happen. Also listed below are expected outputs

and next troubleshooting steps.

## Linecard booted as expected

```
sysadmin-vm:F0_SC0# show platform detail location 0/0
```

```
Platform Information for 0/0
PID : NC6-10X100G-M-P
Description : "NCS 6000 10x100G Multi-Service CXP"
VID/SN : V01
HW Oper State : OPERATIONAL
SW Oper State : OPERATIONAL
Configuration : "NSHUT RST"
HW Version : 1.0
Last Event : HW_EVENT_OK
Last Event Reason : "Initial discovered state:BOOTED (card ok)"
```

Output of Last Event and Last Event Reason show that card is okay. Execute **show reboot history** command to validate if this specific card had issues in the past and if yes, what was the issue.

```
sysadmin-vm:F0_SC0# show reboot-history card location 0/0
```

```
Card Reboot History for 0/0
0
Timestamp "Fri Oct 2 15:15:26 2015"
Reason Code 7
Reason "Install Activate System Reload"
Src Location ""
Src Name INSTALL
Timestamp "Tue Sep 8 18:56:29 2015"
Reason Code 7
Reason "ADMIN CLI RELOAD ROUTER GRACEFUL"
Src Location ""
Src Name "CONFD USER"
```

Aborted: by user

Linecard 0/0 is operational and the reason of the last reload was because of "INSTALL" which essentially means either SMU installation or software upgrade was done. This is expected and thus there were no issues with this card.

## Fabric card unable to boot due to driver error

```
sysadmin-vm:0_RP0# show platform
```

```
Location Card Type HW State SW State Config State
-----
0/FC2 UNKNOWN FAILED N/A NSHUT <-sysadmin-vm:F0_SC0# show
reboot-history card location 0/FC2
```

```
Card Reboot History for 0/FC2
Timestamp "Thu Oct 9 12:10:22 2014"
Reason Code 15 <--
"Board reload as devices not up on Fabric Card"
Src Location 0/FC0
Src Name FAM_AGENT_CALV_DRIVER_SFE B
```

In this example FC card did not boot because the sfe\_driver was not initialized properly.

Execute few more commands to view the reset-history from the CCC point of view. The reboot-history CLI can be used in conjunction with CCC Driver's reset-history CLI to determine the source and reason of card reload.

There could be two options:

**Onboard** - Use this option to look for information if the card went through Warm Reset ( just CPU has been reloaded)

**Onchip** - Use this option to look for information if the card went through Cold Reset ( Card reloaded entirely – Hard reset )

For example:

```

sysadmin-vm:F0_SC0# show controller ccc reset-history on
Possible completions:
  onboard  CCC Reset history in onboard EEPROM detail information
  onchip   On-chip reset history entries since last CCC Cold Reset

sysadmin-vm:F0_SC0# show controller ccc reset-history onchip location 0/0

```

```

*****
***   On Chip Reset History for location 0/0   ***
*****
TimeOfDay      : Tue Oct 20 17:17:40 2015
Uptime         : 18 days 02:01:59 <--
Resets         : 2

```

idx	Reset Source	Reset Command	Reset Time
0	ColdRst	AssrtHR	2015/10/02 15:15:43
1	ColdRst	DeAssrtHR	2015/10/02 15:15:50 --> List reset source as "ColdRst"

```

sysadmin-vm:F0_SC0# show controller ccc reset-history onboard location 0/0

```

```

*****
***   On Board Reset History for location 0/0   ***
*****
Scratch EEPROM Magic      : PON
Scratch EEPROM Version    : 0x00014000
Reset History Magic       : HIST
Number of Resets          : 102 <---

```

idx	Reset Source	Reset Command	Reset Time
0	ColdRst	AssrtHR	1970/01/01 0:00:00 <-- ColdRst
1	ColdRst	DeAssrtHR	1970/01/01 0:00:06
2	ColdRst	AssrtHR	1970/01/01 0:00:00
3	ColdRst	DeAssrtHR	1970/01/01 0:00:06
4	WarmRst	AssrtHR	1970/05/03 7:21:55 <-- WarmRst

<output omitted>

ColdRst can either be initiated by Software or Hardware. WarmRst, is only initiated through Software. Other reset sources can be HRESET\_L, SRESET\_L, Wtchdog, SW\_assgn or plain Rsrvd.

In addition, there are two unique entries per reset operation performed. An AssrtHR operation and a DeAssrtHR operation. This implies that a reset signal has been asserted and then de-asserted therefore the chip will complete the reset.

Note the time stamps of each of these operations. This CLI can be combined with the CCC Inventory Status CLI to determine when the reset occurred and how long the card has been up or down.

Next, check CCC stage that card went through during its reload process. Listed below are different state examples:

### Card that booted correctly from CCC point of view:

```
sysadmin-vm:F0_SC0# show controller ccc event-history brief location 0/0
```

```
CCC Card Event History for: 0/0
```

```
Card Event History as seen by Master (0/RP1)
```

```
Current State: CARD_READY
```

DATE	TIME (UTC)	STATE	EVENT
10/02	15:16:55.234	WAIT_BOOT_IMAGE	ev_boot_ssd_image
10/02	15:16:54.233	BIOS_STARTED	if_wait_ssd_image_booting
10/02	15:16:54.233	CPU_READY	if_bios_started
10/02	15:16:54.231	OIR_INSERT_NOTIF	if_cpu_is_ready
10/02	15:16:54.217	CCC_DRIVER_INIT	if_oir_insert_notif_not_done
10/02	15:16:54.195	PON_POWERED_ON	to_ccc_driver_init
10/02	15:16:54.195	CHECK_CCC_STATUS	if_pon_powered_on
10/02	15:16:54.194	READ_IDPROM	ev_idprom_available
10/02	15:16:53.942	GET_CCC_INFO	ev_get_ccc_info_done
10/02	15:16:53.723	WAIT_ETH_READY	ev_eth_available
10/02	15:16:52.560	CHECK_UBLAZE_BOOT	ev_ublaze_boot_ok
10/02	15:16:52.539	WAIT_CCC_READY	ev_ccc_ready
10/02	15:16:52.537	IDLE	ev_presence_scan

### Card currently in PXE\_BOOTING stage:

```
sysadmin-vm:0_RP0# show controller ccc event-history brief location 0/3
```

```
CCC Card Event History for: 0/3
```

```
Current State: PXE_BOOTING
```

DATE	TIME (UTC)	STATE	EVENT
08/07	19:50:40.607	BIOS_STARTED	if_internal_pxe_booting
08/07	19:50:40.607	WAIT_BIOS_START	ev_bios_started
08/07	19:50:18.605	CPU_READY	if_bios_not_started
08/07	19:50:18.595	CCC_DRIVER_INIT	if_cpu_is_ready
08/07	19:50:18.568	PON_POWERED_ON	to_ccc_driver_init
08/07	19:50:18.568	CHECK_CCC_STATUS	if_pon_powered_on
08/07	19:50:18.567	GET_CCC_INFO	ev_get_ccc_info_done
08/07	19:50:18.550	WAIT_ETH_READY	ev_eth_ready

```

08/07 19:50:18.550 CHECK_UBLAZE_BOOT ev_ublaze_boot_ok
08/07 19:50:18.517 PON_UP_WARM ev_ccc_reset_done
08/07 19:50:12.627 PON_DOWN_WARM ev_pon_up_warm
08/07 19:50:08.239 PON_DOWN_WARM ev_warm_reset_req_ignored
08/07 19:50:07.239 PON_DOWN_WARM ev_warm_reset_req_ignored
08/07 19:50:06.239 PON_DOWN_WARM ev_warm_reset_req_ignored
08/07 19:50:05.239 PON_DOWN_WARM ev_warm_reset_req_ignored
08/07 19:50:04.238 PON_DOWN_WARM ev_warm_reset_req_ignored

```

### Card unable to boot due to chip stuck in GET\_CCC\_INFO:

```
sysadmin-vm:0_RP0# show controller ccc event-history brief location 3/6
```

CCC Card Event History for: 3/6

Card Event History as seen by Master (3/RP0)

Current State: **GET\_CCC\_INFO**

DATE	TIME (UTC)	STATE	EVENT
<b>10/26</b>	<b>23:43:04.559</b>	<b>UBLAZE_NOT_READY</b>	<b>ev_timer_expired</b>
<b>10/26</b>	<b>23:42:34.559</b>	<b>CHECK_UBLAZE_BOOT</b>	<b>ev_timer_expired</b>
10/26	23:42:24.528	WAIT_CCC_READY	ev_ccc_ready
10/26	23:42:21.516	RECOVERY_RESET	ev_timer_expired
10/26	23:42:03.516	CHECK_UBLAZE_BOOT	ev_ublaze_pre_boot_failed
10/26	23:41:52.480	WAIT_CCC_READY	ev_ccc_ready
10/26	23:41:49.468	RECOVERY_RESET	ev_timer_expired
10/26	23:41:32.467	WAIT_CCC_READY	ev_no_fpga_ok_signal
10/26	23:41:29.456	RECOVERY_RESET	ev_timer_expired
10/26	23:41:13.455	WAIT_CCC_READY	ev_no_fpga_ok_signal
10/26	23:41:10.444	RECOVERY_RESET	ev_timer_expired
10/26	23:40:55.444	CHECK_UBLAZE_BOOT	ev_ublaze_pre_boot_failed
10/26	23:40:55.439	WAIT_CCC_READY	ev_ccc_ready
10/26	23:40:52.320	IDLE	ev_presence_scan

### Card unable to boot because of POWER\_UP\_FAILED state:

```
sysadmin-vm:0_RP0# show controller ccc event-history brief location 0/2
```

CCC Card Event History for: 0/2

Current State: **POWER\_UP\_FAILED**

DATE	TIME (UTC)	STATE	EVENT
<b>08/05</b>	<b>14:55:17.449</b>	<b>POWER_UP_FAILED</b>	<b>ev_wdog_timeout</b>
<b>08/05</b>	<b>14:45:31.265</b>	<b>CCC_DRIVER_INIT</b>	<b>if_pwr_up_failed</b>
<b>08/05</b>	<b>14:45:31.260</b>	<b>CHECK_CCC_STATUS</b>	<b>if_pwr_up_failed_again</b>
08/05	14:45:31.258	GET_CCC_INFO	ev_get_ccc_info_done
08/05	14:45:31.223	WAIT_ETH_READY	ev_eth_ready
08/05	14:45:31.157	CHECK_UBLAZE_BOOT	ev_ublaze_boot_ok
08/05	14:45:31.124	PON_UP_WARM	ev_ccc_reset_done
08/05	14:45:17.489	CCC_IN_RESET	ev_pon_up_warm
08/05	14:45:08.921	POWER_UP_FAILED	ev_pon_down_warm
08/05	14:35:07.152	POWER_UP_FAILED	ev_wdog_timeout
08/05	14:25:20.946	CCC_DRIVER_INIT	if_pwr_up_failed
08/05	14:25:20.941	CHECK_CCC_STATUS	if_pwr_up_failed_again
08/05	14:25:20.939	GET_CCC_INFO	ev_get_ccc_info_done
08/05	14:25:20.923	WAIT_ETH_READY	ev_eth_ready
08/05	14:25:20.887	CHECK_UBLAZE_BOOT	ev_ublaze_boot_ok
08/05	14:25:20.830	PON_UP_WARM	ev_ccc_reset_done

Aborted: by user

Using the above command with the "brief" option does not give complete data related to root cause of the issues. For that information replace the **brief** keyword with **detail**.

**Note:** This is the most important CLI when troubleshooting cards not booting on the CCC Level.

Focus on the "Event Desc" and "Reason of the failure" to get better explanation on the failure.

```
sysadmin-vm:F0_SC0# show controller ccc event-history detail location 0/0
```

```
CCC Card Event History for: 0/0
```

```
Card Event History as seen by Master (0/RP1)
```

```
Event buffer info:
```

```
Total number of events recorded: 13
```

```
Number of events available for display: 13
```

```
Current State: CARD_READY
```

```
EVENT #: 12 (record index = 12)
```

```
TIMESTAMP: 2015/10/02 15:16:55.234814 UTC
```

```
STATE: WAIT_BOOT_IMAGE
```

```
EVENT: ev_boot_ssd_image
```

```
EVENT DESC: SSD image is booting
```

```
EVENT #: 11 (record index = 11)
```

```
TIMESTAMP: 2015/10/02 15:16:54.233898 UTC
```

```
STATE: BIOS_STARTED
```

```
EVENT: if_wait_ssd_image_booting
```

```
EVENT #: 10 (record index = 10)
```

```
TIMESTAMP: 2015/10/02 15:16:54.233855 UTC
```

```
STATE: CPU_READY
```

```
EVENT: if_bios_started
```

```
EVENT #: 9 (record index = 9)
```

```
TIMESTAMP: 2015/10/02 15:16:54.231426 UTC
```

```
STATE: OIR_INSERT_NOTIF
```

```
EVENT: if_cpu_is_ready
```

```
EVENT #: 8 (record index = 8)
```

```
TIMESTAMP: 2015/10/02 15:16:54.217351 UTC
```

```
STATE: CCC_DRIVER_INIT
```

```
EVENT: if_oir_insert_notif_not_done
```

```
EVENT #: 7 (record index = 7)
```

```
TIMESTAMP: 2015/10/02 15:16:54.195808 UTC
```

```
STATE: PON_POWERED_ON
```

```
EVENT: to_ccc_driver_init
```

```
EVENT #: 6 (record index = 6)
```

```
TIMESTAMP: 2015/10/02 15:16:54.195786 UTC
```

```
STATE: CHECK_CCC_STATUS
```

Below are example outputs of different scenarios.

**Card that didn't boot due to power problems and got stuck in POWER\_UP\_FAILED:**

Pay attention to ERROR\_INFO to get details about the failure.

```
sysadmin-vm:0_RP0# show controller ccc event-history detail location 0/2
```

```
CCC Card Event History for: 0/2
```

```
Event buffer info:
```

```
Total number of events recorded: 692
```

```
Number of events available for display: 255
```

```
Current State: POWER_UP_FAILED
```

```
EVENT #: 691 (record index = 179)
```

```
TIMESTAMP: 2014/08/05 14:55:17.449979 UTC
```

```
STATE: POWER_UP_FAILED
```

```
EVENT: ev_wdog_timeout
```

```
EVENT DESC: CCC watchdog timeout event
```

```
ERROR INFO: wdog__0 SysAdmin VM Watchdog stage1:0
```

```
<output omitted>
```

## Card that didn't boot and got stuck in CCC\_NOT\_READY (chip problem):

```
sysadmin-vm:0_RP0# show controller ccc event-history detail location 0/FC2
```

```
CCC Card Event History for: 0/FC2
```

```
Event buffer info:
```

```
Total number of events recorded: 2
```

```
Number of events available for display: 2
```

```
Current State: CCC_NOT_READY
```

```
EVENT #: 1 (record index = 1)
```

```
TIMESTAMP: 2014/08/04 14:10:49.891845 UTC
```

```
STATE: WAIT_CCC_READY
```

```
EVENT: ev_ccc_ready_timeout
```

```
EVENT DESC: Timeout waiting for CCC to be ready
```

```
ERROR INFO: CCC READY Timeout - CLOCK_OK signal not being asserted (I/O Expander port0=0xf0, port1=0xff) 8
```

```
<output omitted>
```

There are instances where Cards needs to be removed/re-inserted. For that, CCC component provides OIR-History for given Rack tracking card insertion/removal. Note the keyword "slave" being used. This will give information on the standby RP.

```
sysadmin-vm:0_RP0# show controller ccc oir-history rack 0
```

```
Cards OIR History of rack: 0
```

```
OIR Events as seen by Master (0/RP0)- View from the Active RP
```

DATE	TIME (UTC)	EVENT	LOC	CARD TYPE	SERIAL NO
10/09	16:59:14.280	<b>INSERTED</b>	0/0	NC6-10X100G-M-K	SAL1650UCN9
10/09	16:58:49.064	<b>REMOVED</b>	0/0	NC6-10X100G-M-K	SAL1650UCN9

```
<output omitted> sysadmin-vm:0_RP1# show controller ccc slave oir-history rack 0
```

```
Cards OIR History of rack: 0
```

```
OIR Events as seen by Slave (0/RP1)- <-- View from the standby RP
```

DATE	TIME (UTC)	EVENT	LOC	CARD TYPE	SERIAL NO
11/06	05:54:31.374	DISCOVERED	0/2	NC6-10X100G-M-K	SAD161300XK

<output omitted>

Information must be the same from both RP point of view.

Using the above combination of commands will help to determine the root cause of why the Fabric Card was unable to boot.

## Card stuck in POWEROFF or PRESENT STATE

If the card is stuck in POWEROFF/PRESENT state, it's very likely it went through Multiple Resets and was Powered-OFF by shelf\_mgr.

Issue following commands to determine the root cause of the issue:

1. Collect show tech ctrace from Sysadmin VM
2. show platform detail location <>
3. show reboot-history card location <> (Find out how many times it went through RESET process)
4. show controller ccc event-history detail location <>
5. show controller ccc reset-history onboard location <>

Card can be reset using hw-module reset command and booting process can be observed using rconsole procedure explained later in this document.

```
sysadmin-vm:0_RP1# show controller ccc slave oir-history rack 0
```

Cards OIR History of rack: 0

OIR Events as seen by Slave (0/RP1)- <-- View from the standby RP

DATE	TIME (UTC)	EVENT	LOC	CARD TYPE	SERIAL NO
11/06	05:54:31.374	DISCOVERED	0/2	NC6-10X100G-M-K	SAD161300XK
11/06	05:53:37.442	DISCOVERED	0/6	NC6-10X100G-M-K	SAL1649TN46

<output omitted>

## Card stuck in POWER\_ON STATE

If the card is stuck POWERED\_ON, it means that CCC has turned on the basic power zones that are needed for other Drivers to start their work. It is the responsibility of respective Drivers to move the card to operational state.

The SFE Driver moves FCs to OPERATIONAL state, after it detects and initializes all its internal devices on that card also known as ASICs.

ESD Driver moves SC-SW cards and Scapa's LCs to OPERATIONAL state (almost immediately, nothing to check/initialize unlike SFE Driver).

If Card is stuck in POWERED\_ON state, it means that one of the above drivers had trouble to move the card to Operational state. Problem is seen more often on CPU-less cards. Ex: Fabric Cards or SC's Switch cards (SC-SW).

First step is to check ccc event-history command:

```
sysadmin-vm:F0_SC0# show controller ccc event-history detail location 0/0
```

CCC Card Event History for: 0/0

Card Event History as seen by Master (0/RP1)

Event buffer info:

Total number of events recorded: 13

Number of events available for display: 13

<output ommited>

EVENT #: 7 (record index = 7)

TIMESTAMP: 2015/10/02 15:16:54.195808 UTC

**STATE: PON\_POWERED\_ON**

**EVENT: to\_ccc\_driver\_init**

Next, validate the basic power zones needed:

```
sysadmin-vm:0_RP0# show controller ccc register location 0/RP0 offset 0x4c
```

```
Register      Register
Address       Value
```

```
-----
0x4C          0x3          - zones 0 and 1 OK sysadmin-vm:0_RP0# show controller ccc register
location 0/RP0 offset 0x50
```

```
Register      Register
Address       Value
```

```
-----
0x50          0x3          - zone 0 and 1 Enabled
```

```
sysadmin-vm:0_RP0# show controller ccc power detail location 0/RP0
```

Power detail : Zone information for 0/RP0:

```
-----
| Power Zone | Power Status | Power Contrl | Power Fault |
-----
| 0          | OK          | SET          | --          | - Power Status OK
| 1          | OK          | SET          | --          | - Power Status OK sysadmin-
```

```
vm:F0_SC0# show controller ccc i2c-dev ioexpander location 0/0
```

CCC IO Expander information for location: 0/0

Port 0: 0x3e

Port Bit I/O Val Bit Name

```
-----
P0 0 0 0 Power Cycle
P0 1 I 1 FPGA OK
P0 2 I 1 uBlaze OK
P0 3 I 1 Clock OK
P0 4 I 1 Core Volt OK
P0 5 I 1 OTH Volt0 OK
P0 6 I 0 OTH Volt1 NOT OK
P0 7 I 0 OTH Volt2 NOT OK
```

Port 1: 0x3

Port Bit I/O Val Bit Name

```
-----
P1 0 I 1 FPGA INIT OK
```

If above valiation do not lead to a root cause then next step would be to open a TAC Service Request.

## Card stuck in SW\_INACTIVE STATE

```
sysadmin-vm:0_RP0# show platform
```

Location	Card Type	HW State	SW State	Config State
0/1	P-L-10X100G-F-P	POWERED_OFF	<b>SW_INACTIVE</b>	SHUT
0/RP0	P-L-RP	OPERATIONAL	OPERATIONAL	NSHUT
0/RP1	P-L-RP	OPERATIONAL	OPERATIONAL	NSHUT
0/FC0	P-L-FC-S	OPERATIONAL	N/A	NSHUT
0/FC1	P-L-FC-S	OPERATIONAL	N/A	NSHUT
0/FT0	PANINI-SIM-FT	OPERATIONAL	N/A	NSHUT
0/FT1	PANINI-SIM-FT	OPERATIONAL	N/A	NSHUT

Possible failure reasons are:

- Host OS not booting due to SSD access issue.
- Host OS booting stuck due to HW issue.
- SysAdmin VM failed to get spawned.
- Control Ethernet connection issues:
- Card MAC/IP not programmed due to SW failure
- Ethernet switch not getting programmed correctly by CCC Power-On interpreter.
- ESD Color Switch image failed to get programmed in the CCC SPI flash.

### Scenario 1: SW\_EVENT\_FAILURE: SW\_EVENT\_ADMIN\_VM\_FAILURE reported by shelf\_mgr syslogs

```
sysadmin-vm:0_RP0# show platform
```

Location	Card Type	HW State	SW State	Config State
0/1	PROTO-CXP-2XPITA	OPERATIONAL	OPERATIONAL	NSHUT
0/RP0	NC6-RP	OPERATIONAL	OPERATIONAL	NSHUT
<b>0/RP1</b>	<b>NC6-RP</b>	<b>OPERATIONAL</b>	<b>SW_INACTIVE</b>	<b>NSHUT</b>
0/FC0	NC6-FC-MC	OPERATIONAL	N/A	NSHUT
0/CI0	P-L-CRFT	OPERATIONAL	N/A	NSHUT
0/FT0	P-L-FANTRAY	OPERATIONAL	N/A	NSHUT
0/FT1	P-L-FANTRAY	OPERATIONAL	N/A	NSHUT

There could be several different reason's why RP1 could not boot. The easiest way to find out the issue is to rconsole on the RP and check the logs on(Refer to rconsole procedure on the bottom of this document).

### Scenario 2: LC stuck in FAILED state, last event: HW\_EVENT\_FAILURE fail\_code=LC\_POWER\_MAIN\_FAULT

Ensure the HW state shows FAILED and SW State shows SW\_INACTIVE:

```
sysadmin-vm:0_RP0# show platform location 0/1
```

Location	Card Type	HW State	SW State	Config State
0/1	NC6-60X10GE-M-S	<b>FAILED</b>	<b>SW_INACTIVE</b>	NSHUT

Execute below command and check "Last Even Reason":

```
sysadmin-vm:0_RP0# show platform detail location 0/1
```

Platform Information for 0/1

```
PID : NC6-60X10GE-M-S
Description : "NCS 6000 60x10G Multi-Service SFP+"
VID/SN : V01
HW Oper State : FAILED
SW Oper State : SW_INACTIVE
Configuration : "NSHUT_RST"
HW Version : 0.6
Last Event : HW_EVENT_FAILURE
Last Event Reason : "pon exit <-- UP_WARM_RESET cnt=123 fail_code=LC_POWER_MAIN_FAULT"
```

```
sysadmin-vm:0_RP0#
```

Filter syslog for the affected card to check log messages:

```
0/RP0/ADMIN0:Jun 21 00:33:13.487 : cm[1795]: %ROUTING-TOPO-5-OIR_ACTION : OIR card failed
having serial number: SAD173501R7.
0/RP0/ADMIN0:Jun 21 00:33:13.528 : shelf_mgr[1818]: %INFRA-SHELF_MGR-5-CARD_INSERTION :
Location: 0/1, Serial #: SAD173501R7
0/RP0/ADMIN0:Jun 21 00:33:13.528 : shelf_mgr[1818]: %INFRA-SHELF_MGR-6-HW_EVENT : Rcvd HW event
HW_EVENT_FAILURE, event_reason_str 'Initial discovery FAIL: EXIT0, power request on , but not
finish ccc-pon startup. power_control 0x00000001' for card 0/1
0/RP0/ADMIN0:Jun 21 00:33:13.530 : shelf_mgr[1818]: %INFRA-SHELF_MGR-3-CARD_HW_FAILED : Card:
0/1 hardware state going to FAILED
0/RP0/ADMIN0:Jun 21 00:34:06.734 : shelf_mgr[1818]: %INFRA-SHELF_MGR-6-HW_EVENT : Rcvd HW event
HW_EVENT_RESET, event_reason_str 'pon enter --> DOWN_WARM_RESET cnt=3! ' for card 0/1
0/RP0/ADMIN0:Jun 21 00:34:15.987 : shelf_mgr[1818]: %INFRA-SHELF_MGR-6-HW_EVENT : Rcvd HW event
HW_EVENT_POWERED_OFF, event_reason_str 'CCC Warm Reset #8' for card 0/1
0/RP0/ADMIN0:Jun 21 00:34:21.419 : cm[1795]: %ROUTING-TOPO-5-OIR_ACTION : OIR card failed having
serial number: SAD173501R7.
0/RP0/ADMIN0:Jun 21 00:34:21.459 : shelf_mgr[1818]: %INFRA-SHELF_MGR-3-CARD_HW_FAILED : Card:
0/1 hardware state going to FAILED
0/RP0/ADMIN0:Jun 21 00:34:21.459 : shelf_mgr[1818]: %INFRA-SHELF_MGR-6-HW_EVENT : Rcvd HW event
HW_EVENT_FAILURE, event_reason_str 'pon exit <-- UP_WARM_RESET cnt=4
fail_code=LC_POWER_MAIN_FAULT' for card 0/1
```

Execute **show reboot history** command and verify if there is huge time difference between AssrtHR and DeAssrtHR messages. This is probably because there is internal connectivity issue between the VM's.

```
sysadmin-vm:0_RP0#show reboot-history card location 0/1
```

Reset history example: every 20 mins for 2 hours before it recovered:

```
33 0 WarmRst DeAssrtHR 0x00000F32 0x53A4D367 Sat Jun 21 00:35:51 2014
34 0 WarmRst AssrtHR 0x00000F10 0x53A4D81D Sat Jun 21 00:55:57 2014

35 0 WarmRst DeAssrtHR 0x00000F32 0x53A4D821 Sat Jun 21 00:56:01 2014
36 0 WarmRst AssrtHR 0x00000F10 0x53A4DCD7 Sat Jun 21 01:16:07 2014
```

### Card stuck in UNKOWN STATE

When the Card reports Admin state as UNKNOWN, most likely CCC could not read the IDPROM from the board therefore card will not be able to complete the boot. For such instances, execute the following commands on the given location:

```
RP/0/RP0/CPU0:A41-PE1#show platform
```

```
Node name Node type Node state Admin state Config state
```

```

-----
0/RP1          NC6-RP          OPERATIONAL    UNKNOWN
0/FC1          NC6-FC          OPERATIONAL    UNKNOWN sysadmin-vm:F0_SC0# show
controller ccc event-history brief location 0/0

```

CCC Card Event History for: 0/0

Card Event History as seen by Master (0/RP1)  
 Current State: CARD\_READY

DATE	TIME (UTC)	STATE	EVENT
10/02	15:16:54.194	READ_IDPROM	<b>ev_idprom_available</b>
10/02	15:16:53.942	GET_CCC_INFO	ev_get_ccc_info_done
10/02	15:16:53.723	WAIT_ETH_READY	ev_eth_available
10/02	15:16:52.560	CHECK_UBLAZE_BOOT	ev_ublaze_boot_ok
10/02	15:16:52.539	WAIT_CCC_READY	ev_ccc_ready
10/02	15:16:52.537	IDLE	ev_presence_scan

Next step would be to verify if CCC is running on the LC during bootup process using rconsole:

### 1. Collect the process ID in SysadminVM:

```
sysadmin-vm:F0_SC0# show processes ccc_driver location 0/0
```

```

-----
                PID: 2525
    Executable path: /opt/cisco/calvados/packages/ncs6k-sysadmin-boot-5.2.4.CSCut24295
.all-1.0.0/sbin/ccc_driver
    Instance #: 0
    Respawn: ON
    Respawn count: 1
    Max. spawns per 4 mins: 4
    Last started: 10/02/2015 15:17:23.000
    Process state: Run
    startup_path: /opt/cisco/calvados/packages/ncs6k-sysadmin-boot-5.2.4.CSCut24295
.all-1.0.0/etc/startup/ccc_driver.startup
    Ready: 5s

```

### 2. Rconsole to the LC using the below commands:

```
sysadmin-vm:F0_SC0# attach location 0/RP0
```

```

[sysadmin-vm:0_RP0:~]$ exec chvrf 2 bash
[sysadmin-vm:0_RP0:~]$ chvrf 0 bash
[sysadmin-vm:0_RP0:~]$ /opt/cisco/calvados/sbin/rconsole -l 0/0
Connecting to location 0/0 (backplane-slotid 16, console 0)
Escape sequence is "end"
Waiting for card info from CCC-driver for slot 16
Got card info from CCC-driver for slot 16
IOS Build Date : 04/22/2015 by lchinnad
System Memory Speed : 1334 MHz
Processor Type : Intel(R) Xeon(R) CPU E5-2418L @ 2.00GHz

```

Press F12 to goto Boot Manager..

```

Booting System Host OS..
Waiting For CCC Valid Time of Day..
Waiting For CCC Valid Time of Day..
CCC Time: Fri Oct 2 15:16:54 2015

```

```
GNU GRUB version 2.00
Press F2 to goto grub Menu..
Booting from Disk..
Loading Kernel..
Loading initrd..
[ 1.949229] i8042: No controller found
Starting udev: [ OK ]
Switching to new root and running init.
Starting udev: [ OK ]
Actual changes:
large-receive-offload: off [requested on]
ntuple-filters: on
Setting hostname host: [ OK ]
Checking filesystems:[ OK ]
Entering non-interactive startup
Bringing up loopback interface: [ OK ]
Bringing up interface eth0: Device eth0 does not seem to be present, delaying initialization.
[FAILED]
Starting system logger: [ OK ]
Starting kernel logger: [ OK ]
Starting kdump:[ OK ]
Starting system message bus: [ OK ]
Starting smartd: [ OK ]
Generating SSH1 RSA host key: [ OK ]
Generating SSH2 RSA host key: [ OK ]
Generating SSH2 DSA host key: [ OK ]
Starting sshd: [ OK ]
Starting xinetd: [ OK ]
Starting crond: [ OK ]
Starting libvirtd daemon: [ OK ]
Starting NCS6k programs for LC on hostos: [ OK ]
mcelog start/running, process 2637
Creating default host password file
serial (/dev/ttyserial (/dev/ttyS1) start/running, process 2649
```

```
host login: root
Password:
[host:~]$
[host:~]$
[host:~]$ telnet 0 50001 <-- to get to Calvados
Trying 0.0.0.0...
Connected to 0.
Escape character is '^'.
```

```
sysadmin-vm:0_0 login:
sysadmin-vm:0_0 login: root
Password:
```

```
[sysadmin-vm:0_0:~]$ pgrep ccc <- use pgrep to check if the process is running
2525
[sysadmin-vm:0_0:~]$ exit
logout
```

## Using RCONSOLE tool

When cards are unable to boot, The NCS6008 provides built in feature which gives the capability to perform remote console on the card and see the reason why the card is stuck and unable to boot. This feature is called RCONSOLE and below is an example of its usage.

Procedure to Rconsole on specific LC:

1. Go to SysadminVM
2. Attach to active RP
3. Change to global VRF chvrf 0 bash
4. Execute /opt/cisco/calvados/sbin/rconsole -l (Linecard)

Example:

```
RP/1/RP1/CPU0:6008-B#admin
sysadmin-vm:F0_SC0#
  sysadmin-vm:F0_SC0# attach location 0/RP0 <-- You must be connected to the RP's to be able to
rconsole
Tue Oct 20 18:23:54.740 UTC
[sysadmin-vm:0_RP0:~]$ exec chvrf 2 bash
[sysadmin-vm:0_RP0:~]$ chvrf 0 bash
[sysadmin-vm:0_RP0:~]$ /opt/cisco/calvados/sbin/rconsole -l 0/0 B This is LC 0/0
Connecting to location 0/0 (backplane-slotid 16, console 0)
Escape sequence is "end"
Waiting for card info from CCC-driver for slot 16
```

This procedure is widely used by TAC to determine the current state of the card and check where its stuck.

## List of commands to be collected before opening TAC case

### XR VM:

```
Show platform
show install active
Show version
Show reboot history location 0/0/cpu0
dir misc/disk1
show cli history detail
Show log
show tech-support npu
```

### SysAdmin VM:

```
show platform detail
show platform slices
show install active
show sdr default-sdr reboot-history
show reboot-history card location <>
show controller ccc reset-history onbo loc <>
show controller ccc reset-history onch loc <>
show controller ccc event-history detail location <>
show tech-support ccc
show tech-support Hbloss
show tech-support slice_manager
show tech-support ctrace
show tech-support sdr_mgr
show log
```