

Runbook of Foximan L11

- 1. Running L11
 - 1.1. Prerequisites
 - 1.2. Command Cheatsheet
 - 1.2.1. Foximan Commands
 - 1.2.2. Screen Session Commands
 - 1.2.3. Burninator Ops Commands
 - 1.3. Step-by-step Guide
 - 1.3.1. Lookup the Rack Serial Number and Generate the Rack Info
 - 1.3.2. Validate the Rack Info and Connectivity
 - 1.3.3. Run L11 Precheck
 - 1.3.4. Run Partnerdiag
 - 1.3.5. Reseating a Failing Compute/Switch Tray
 - 1.3.6. Replacing a Failing Compute/Switch Tray
 - 1.3.7. Handing the Rack back to the Customer
 - 1.3.8. Running nvlmapper Tool
- 2. Troubleshooting
- 3. Appendix
 - 3.1. Script for start_l11

The following is the Foximan L11 test runbook for ABL. **This is specific to L11 rack level testing.** For testing individual compute trays, please see the [Runbook of Foximan repair endpoint](#).

1. Running L11

1.1. Prerequisites

Following the Prerequisites in [Runbook of Foximan repair endpoint](#) for accessing both the Diag Server and the burninatorOps tool.

1.2. Command Cheatsheet

For the commands below, set the environment variables **USERNAME** to your name and **SERIAL** to the rack serial number.

1.2.1. Foximan Commands

Description	Command	Action on Failure
Validate credentials and serial numbers for all compute and switch trays	<code>start_l11 \${USERNAME} \${SERIAL} rack_validate</code>	Verify serial number/ip addresses for mismatch Reassign back to customer for unknown credentials
Run a fast version of prechecks to get the Rack booted in the correct order and into the host OS with NMX controller active. This also validates potential fabric issues after bootup.	<code>start_l11 \${USERNAME} \${SERIAL} roma_l11_pre</code>	Check error. May need to move to full OCI precheck to repave all trays.
Only run this if the fast version above fails OR if a tray is replaced in the rack Runs a long (~2-3 hours) version of precheck, where all trays are repaved with factory defaults and new firmware. This is a superset of the above	<code>start_l11 \${USERNAME} \${SERIAL} oci_fvt_pre</code>	Check logs for error. See Troubleshooting below.
Run a full L11 NVIDIA Partnerdiag suite of tests	<code>start_l11 \${USERNAME} \${SERIAL} nv</code>	Check logs for error. See Troubleshooting below.
Run this after reseating any tray in the rack. This will force the fabric to reset and (hopefully) clear any negotiation errors.	<code>start_l11 \${USERNAME} \${SERIAL} fabric_reset</code>	Check logs for error. See Troubleshooting below.
Run this before handing back the rack to the customer. This will stop the host OS that we use for testing.	<code>start_l11 \${USERNAME} \${SERIAL} rack_clean</code>	Check logs for error. See Troubleshooting below.
Run the nvlmapper tool to generate the mapping of NV Link issues.	<code>start_l11 \${USERNAME} \${SERIAL} nvlmapper</code>	NMX Controller is probably not running, so make sure that is enabled.

1.2.2. Screen Session Commands

Description	Command
Attach to screen session for the rack	<code>screen -r \${USERNAME}_\${SERIAL}</code>

Tail the screen session logs	<code>tail -f screen_logs/\${USERNAME}/\${USERNAME}_\${SERIAL}.log</code>
------------------------------	---

1.2.3. Burninator Ops Commands

Description	Command
Register for Access (Daily)	<code>burninatorOps register -r us-abilene-1 -p abl -ac abl -e https://rmcromaabl.us-abilene-1.oci.oraclecloud.com</code>
Get Rack Info	<code>burninatorOps automate ssh-config-setup -j jump.enclave.rmcromaabl.abl -k ~/.ssh/id_pxe_instance_rsa -rs \${SERIAL}</code>
Copy Rack Info to the Diag Server	<code>scp ~/rack_info/\${SERIAL}.txt ubuntu@rmc-abl15-1949-gpu18:~/rack_info</code>

1.3. Step-by-step Guide

For the instructions below, set the environment variables **USERNAME** to your name and **SERIAL** to the rack serial number.

1.3.1. Lookup the Rack Serial Number and Generate the Rack Info

Use Burninator Ops tool to generate the rack info for your rack under test.

```
$ burninatorOps register -r us-abilene-1 -p abl -ac abl -e https://rmcromaabl.us-abilene-1.oci.oraclecloud.com
$ burninatorOps automate ssh-config-setup -j jump.enclave.rmcromaabl.abl -k ~/.ssh/id_pxe_instance_rsa -rs
${SERIAL}
```

This will generate a file under `~/rack_info/${SERIAL}.txt` that contains all compute tray and switch tray serial numbers and IP addresses. That file needs to get copied to the Diag server.

```
$ scp ~/rack_info/${SERIAL}.txt ubuntu@rmc-abl15-1949-gpu18:~/rack_info
```

1.3.2. Validate the Rack Info and Connectivity

 Always validate the rack first when starting tests AND after tray replacement

Connect to the Diag server and run a validation to check the compute and switch trays for connectivity and if the correct serial number is associated to that IP address.

```
$ start_l11 ${USERNAME} ${SERIAL} rack_validate
```

This will run the command in a screen session in the background with all logs output to `~/screen_logs/${USERNAME}/${USERNAME}_${SERIAL}`. Note: if your screen session is already active, it will reuse the same session. I recommend watching the log file to see status.

```
$ tail -f screen_logs/${USERNAME}/${USERNAME}_${SERIAL}.log
```

You can also connect to the screen session using:

```
$ screen -r ${USERNAME}_${SERIAL}
```

1.3.3. Run L11 Precheck

The prechecks will get the rack ready for running partnerdiag. There is a fast precheck (**roma_l11_pre**) and a slower full precheck (**oci_fvt_pre**). The fast precheck will not factory reset nor will it reflash the PLDM firmware on the compute trays. It will upgrade the switch trays if required. Use the slower precheck after replacing a compute/switch tray OR if the fast precheck fails.

```
$ start_l11 ${USERNAME} ${SERIAL} roma_l11_pre
```

OR

```
$ start_l1l ${USERNAME} ${SERIAL} oci_fvt_pre
```

1.3.4. Run Partnerdiag

The NVIDIA L11 Partnerdiag suite of tests is run after prechecks pass.

```
$ start_l1l ${USERNAME} ${SERIAL} nv
```

Once partner diagnostics suite starts running, you will see a log message on the screen that looks like:

```
2025-09-23 20:42:41,680 [INFO] :: Found Switch log file at /mnt/storage/PartnerDiag/629-24972-4975-FLD-43744-
rev2/2524XN8016/0/2524XN8016_switch_node.log
```

The above is for the NV Switch partnerdiag suite of tests. There will be a different file for Compute tray partnerdiag output. To watch the progress of the Partner Diagnostic run, you can watch that log file.

```
$ tail -f /mnt/storage/PartnerDiag/629-24972-4975-FLD-43744-rev2/2524XN8016/0/2524XN8016_switch_node.log
```

The failures in partnerdiag will get listed in that log above. For a pass, you will see a message like:

```
#####      ####      #####      #####
#####      #####      #####      #####
##  ##  ##  ##  ##  ##  #  ##  #
##  ##  ##  ##  ##  ##  ##  ##
#####      #####      #####      #####
#####      #####      ##      ##
##  ##  ##  ##  #  ##  #  ##
##  ##  ##  ##  #####  #####
##  ##  ##  ##  #####  #####
```

Final Result: PASS

Failures will show a big failure message with a summary of what failed:

Exit Code	Virtual Id	Test	Subtest	Component	Component
Id					
Notes					
=====					
=====					
=====					
MODS-00000000140 ThermalSteadyStateLoop1 powersync nmx-telemetry CPU, GPU SWITCH_NODE_8, MF0:abl15-2203-nvswitch8:N5200_LD/U2, NVLINK 30, Chassis Slot 16, Tray 7, Remote Device GB100 Nvidia Technologies, Remote Port 16, Remote GUID 0x40cc89e7ae3545dc FecEffectiveBer Bit Error Rate Found 6e-08, exceeded threshold 1e-09					

See the troubleshooting guide below when triaging partnerdiag failures.

1.3.5. Reseating a Failing Compute/Switch Tray

When identifying a bad compute or switch tray, some issues will remediate after a reseating (unplug and plugging in the tray). After reseating a compute tray, **ALL** switch trays should also be reseated. After reseating run a fabric reset to clear any fabric issues.

```
$ start_l1l ${USERNAME} ${SERIAL} fabric_reset
```

After a fabric reset, you can move directly to running partnerdiag again. Only re-run prechecks if a fabric reset fails.

1.3.6. Replacing a Failing Compute/Switch Tray

When replacing a failing compute or switch tray, you will need to go back to the Precheck step and run a **FULL oci_fvt_pre** precheck. This will perform a full repave of the rack, to ensure that all trays are running the exact same firmware after replace.

```
start_l11 ${USERNAME} ${SERIAL} oci_fvt_pre
```

Once this completes, move to running partnerdiag.

1.3.7. Handing the Rack back to the Customer

Before handing the rack back to the customer, the rack needs to have the test host OS stopped. This will prepare the rack for the customer.

```
start_l11 ${USERNAME} ${SERIAL} rack_clean
```

1.3.8. Running nvlmapper Tool



The nvlmapper tool requires the NMX controller to be running (and I believe it also requires the Prometheus endpoint for telemetry). These are automatically started as part of prechecks or fabric reset.

The nvlmapper tool is used to generate an Excel xlsx file showing all problematic links/pins, along with a summary sheet of all acp errors. To generate this, just call:

```
start_l11 ${USERNAME} ${SERIAL} nvlmapper
```

On success, the logs will output the following: `

```
NVL Mapper generated output to '/var/log/foximan/runs/2025-09-24_22-09-54_208089_0  
/Rack0_NVLINK_CC_PORTMAP_e2e_rack4_bianca_72x1.xlsx'
```

Copy that file off the Diag server and open in Excel to visualize the errors.

2. Troubleshooting

Add errors here.

3. Appendix

3.1. Script for start_l11

```
#!/bin/bash
USER=$1
SERIAL=$2
FOXIMAN_TEST=$3

if [[ -z "${USER}" ]]; then
    echo "Enter Username:"
    read USER
fi
if [[ -z "${SERIAL}" ]]; then
    echo "Enter Compute Tray Serial Number:"
    read SERIAL
fi
if [[ -z "${FOXIMAN_TEST}" ]]; then
    echo "Enter Test (oci_fvt_pre, nv, fabric_reset, clean):"
    read FOXIMAN_TEST
fi
echo

CREDENTIALS="--credentials ${HOME}/.foximan/111_credentials"
SCREEN_SESSION="${USER}_${SERIAL}"

mkdir -p "${HOME}/screen_logs/${USER}"

screen -ls | grep -q "\.${SCREEN_SESSION}"
if [[ $? -eq 1 ]]; then
    screen -dmS "${SCREEN_SESSION}" -L -logfile "${HOME}/screen_logs/${USER}/${SCREEN_SESSION}.log"
fi
screen -S "${SCREEN_SESSION}" -X stuff "sudo foximan_emmanuel ${FOXIMAN_TEST} ${CREDENTIALS} --rack_info ${HOME}/rack_info/${SERIAL}.txt ${SERIAL}\n"

echo "Foximan started for serial ${SERIAL}"
echo
echo "Attach to screen session using: screen -r ${SCREEN_SESSION}"
```