

Diagnosis and Serviceability

This chapter contains the following topics:

- About Diagnosis and Serviceability, on page 1
- Show Commands, on page 1
- Debug Logs, on page 3
- Diagnosis Suggestions, on page 6
- Revision History, on page 7

About Diagnosis and Serviceability

Cisco NX-OS supports Model-Driven Programmability (MDP) through a range of different protocol interfaces, such as Netconf, Restconf, gNMI/gNOI, and Telemetry. In fact, these interfaces operate around the common underlying YANG and DME/CLI infrastructure. The user can diagnose the behavior through a common collection of utilities.

Show Commands

This section lists the commonly used show commands that you can use to verify the running state of the switch.

Table 1: Show Commands - Diagnosis and Serviceability

Item	Command	Usage
netconf	**show netconf nxsdk event-history { msgs | cli | internal libnxsdk | all }**	Display specific event history
	show netconf nxsdk event-history component { agent | mtx }	Display component event history
	show netconf internal mtx	Display MTX infra information
	show running-config netconf	Display netconf config
	show netconf nxsdk event-history {**events** | **errors**}	Display event history
	show tech-support netconf	Collect netconf tech-support
restconf	**show running-config** | **grep restconf**	Display restconf config
	show retconf nxsdk event-history {**events** | **errors**}	Display event history
	show restconf nxsdk event-history { msgs | cli | internal libnxsdk | all }	Display specific event history
	show restconf nxsdk event-history component { agent | mtx }	Display component event history
	show restconf internal mtx	Display MTX infra information
	show tech-support restconf	Collect restconf tech-support
grpc	**show running-config grpc**	Display grpc config
	show grpc nxsdk event-history {**events** | **errors**}	Display event history
	show tech-support grpc	Collect grpc tech-support
	show grpc nxsdk event-history { msgs | cli | internal libnxsdk | all }	Display specific event history
	show grpc nxsdk event-history component { agent | mtx | telemetry }	Display component event history
	show grpc internal mtx	Display MTX infra information

Item	Command	Usage
gnmi	**show grpc gnmi service statistics**	Verify grpc server state
	show grpc gnmi rpc [**all**] { **summary** | **detail** }	List current/history gNMI subscription
	show grpc gnmi transactions	List gNMI Get/Set
gnoi	**show grpc gnoi service statistics**	Verify grpc server state
openconfig	**show running-config openconfig**	Display openconfig config
	show openconfig nxsdk event-history {**events** | **errors**}	Display event history
dme	**show system internal dme transaction history**	Verify the DME transaction
	show tech-support dme	Collect DME tech-support

Debug Logs

This section describes how to enable and collect the debug logs.

Programmability Agent Logs

For Netconf, Restconf, and gRPC agents, you can collect the logs in the following ways:

- **Show commands**

  This is a straight-forward way to view/check the agent event. These commands are useful to see how the agents interact with the client connections.

```
show netconf nxsdk event-history { events | errors }
show netconf nxsdk event-history component { agent events | mtx events }
show restconf nxsdk event-history { events | errors }
show restconf nxsdk event-history component { agent events | mtx events }
show grpc nxsdk event-history { events | errors }
show grpc nxsdk event-history component { agent events | mtx events | telemetry events
 }
```

- **Log files**

  If you prefer to check the longer history, or even the logs after disabling the agents, then see the log files stored under the **/volatile** directory. The user needs the permission to access the switch bash shell.

```
/volatile/netconf-internal-log
             grpc-internal-log
             restconf-internal-log
```

YANG Infra Logs

The YANG infra debug/trace level logs and file logging are disabled by default. The user can enable file logging. Once file logging is enabled, the YANG infra logs are saved in /volatile directory.

The user needs the permission to access the switch bash shell.

```
debug netconf mtx file
debug restconf mtx file
debug grpc mtx file

/volatile/mtx-internal.netconf.log
              mtx-internal.grpc.log
              mtx-internal.restconf.log
```

DME Logs

The DME infra logs are saved in the **/nxos/dme_logs** directory. The user needs the permission to access the switch bash shell. See https://developer.cisco.com/site/cisco-nexus-nx-api-references/.

```
/nxos/dme_logs/svc_ifc_policyelem.<pid>.log
```

XOS Logs

The XOS logs exposed by tmtrace.bin commands.

The user needs the permission to access the switch bash shell.

```
tmtrace.bin -d cmi-errors
tmtrace.bin -d cmi-events
tmtrace.bin -d cmi-debug
tmtrace.bin -d cmi-msg
```

Change the Log Configuration Using CLI

CLIs are available to change the above logging config dynamically without restarting the process. These commands are per agent EXEC commands and no changes to the config file, and thus can be changed without impacting the current operations.

SUMMARY STEPS

1. **[no] debug grpc mtx enable-all**
2. **[no] debug grpc mtx level <level>**
3. **[no] debug grpc mtx item <item>**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**[no] debug grpc mtx enable-all**	This is a convenient cli to enable all logs.
Step 2	**[no] debug grpc mtx level <level>**	Toggle the logging level: error, warning, info, debug.

	Command or Action	Purpose
	Example: `switch# debug grpc mtx level info`	The default level is **info**.
Step 3	**[no] debug grpc mtx item <item>** **Example:** `switch# debug grpc mtx item MTX-EvtMgr`	Toggle the logging for specific item. This is a free form string, and use **show grpc internal mtx debug** to see the available items.

Example

The below show cli would display the current logging configuration.

show grpc internal mtx debug

```
Example:
  switch# show grpc internal mtx debug
  Log enabled  : 1
  File enabled : 0
  All active   : 0
  Log Level    : Info
  Log items    :
    *                              : 0
    DB                             : 0
    DtxUserFunc                    : 0
    INST-ADAPTER                   : 0
    INST-ADAPTER-GNMI              : 0
    INST-ADAPTER-GNOI              : 0
    INST-ADAPTER-GRPC              : 1
    INST-ADAPTER-NC                : 1
    INST-ADAPTER-RC                : 1
    INST-ADAPTER-TM                : 0
    INST-MTX-API                   : 1
    LIBUTILS                       : 1
    MTX-API                        : 1
    MTX-ActionMgr                  : 0
    MTX-Coder                      : 0
    MTX-Dy-EvtMgr                  : 0
    MTX-EvtMgr                     : 0
    MTX-RbacMgr                    : 0
    MTXEXPR                        : 0
    MTXItem                        : 0
    MTXNetConfMessage              : 0
    MTXOperation                   : 0
    MTXRestConfMessage             : 0
    MTXgNMIMessage                 : 0
    Model-*                        : 1
    Model-Cisco-NX-OS-device       : 1
    Model-openconfig-bgp           : 0
    PVSH                           : 0
    RPC                            : 0
    SYSTEM                         : 1
    TM-ADPT                        : 0
    TM-ADPT-JSON                   : 0
```

Diagnosis Suggestions

This section provides a few steps to triage frequently seen issues.

Connection Issues

If the user's programming client cannot connect to the switch, then check the following:

- Check whether the feature is enabled by checking the running configuration.
- Check individual agent's show command to confirm that the server is running.
- Check the ip / port to confirm the connectivity is not restricted by firewall, etc.
- Check the client sends the correct user/password.
- If cert-based authentication is used, check that the trustpoint has been properly configured to the switch, and the client certification matches and has not expired.

Native Device Yang

If there is an issue with the native openconfig YANG releated to read/write operations, then check the following:

- For "write" operations, check the DME transaction to see the failure details.
- Send equivalent DME REST request, to confirm whether it has the same issue.

OpenConfig Yang

If there is issue read/write the native openconfig YANG, then check the following:

- Check whether **feature openconfig** is enabled.
- Check the published YANG and deviation to confirm the support status.
- For **write** operations, check the DME transaction to see the failure details.

Telemetry

Telemetry is used to collect YANG and other data sources through the "feature telemetry" configuration. Telemetry is also used for gNMI subscribe via "feature grpc". Troubleshooting steps are different depending on the usage scenario.

Debug Logs

Debug logs can be viewed through:

- **show telemetry internal event history { errors | events }**
- **show grpc nxsdk event-history { events | errors }**

Data / Event Collection Issues:

Check show command for failed or skipped collections.

- **show telemetry data collector detail**

- **show telemetry event collector {errors | stats}**
- **show grpc internal gnmi subscription statistics**

Collection time or size issues:

Check collection sizes and times via following show commands:

- **show telemetry control database**
- **show grpc internal gnmi rpc subscription-data**

Transport Issues:

Check for transport issues with following show command. Note that transport issues only impact **feature telemetry** scenario.

- **show telemetry transport <num> stats | errros**

Revision History

Table 2: Revision History - Diagnosis and Serviceability

Release	Description
10.4(2)	New CLI commands to modify logging parameters
10.5(3)	Logging 2.0 support. Add new component event-history commands. Support feature dynamic event-history commands