



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 2](#)
- [Diagnosis Suggestions, on page 10](#)

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 running-config netconf	Display netconf config
	show netconf nxsdk event-history {events errors}	Display event history
	show tech-support netconf	Collect netconf tech-support
	show netconf internal details	Verify internal state
	show netconf internal tls service	Verify TLS server state
	show netconf internal tls session [all] { summary detail }	List current/history TLS sessions

Item	Command	Usage
restconf	show running-config grep restconf	Display restconf config
	show netconf nxsdk event-history {events errors}	Display event history
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
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
	show grpc internal gnmi subscription...	Display internal subscription data
	show grpc internal gnmi mtm {sessions statistics subscriptions}	Display internal infra logs
gnoi	show grpc gnoi service statistics	Verify grpc server state
	show grpc internal gnoi rpc [all] {summary detail}	List current/history gNOI connections
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. This log is in-memory log, and thus it could only keep a relatively short history.

```
show netconf nxsdk event-history {events | errors}
show restconf nxsdk event-history {events | errors}
show grpc nxsdk event-history {events | errors}
```

- **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 logs are saved in the **/volatile** directory. The user needs the permission to access the switch bash shell. In Cisco NX-OS, Bash is accessible from user accounts that are associated with the Cisco NX-OS dev-ops role or the Cisco NX-OS network-admin role.

```
/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
```

Change the Log Configuration

Cisco NX-OS enables very limited logs by default due to the performance consideration.

The user can change the verbosity by editing **/opt/mtx/conf/mtxlogger.cfg**.

The configuration file has the following structure:

```
<config name="nxos-device-mgmt">
  <container name="mgmtConf">
    <container name="logging">
      <leaf name="enabled" type="boolean" default="false"></leaf>
      <leaf name="allActive" type="boolean" default="false"></leaf>
      <container name="format">
        <leaf name="content" type="string" default="$DATETIME$ $COMPONENTID$ $TYPES$ :
$MSG$"></leaf>
        <container name="componentID">
          <leaf name="enabled" type="boolean" default="true"></leaf>
        </container>
        <container name="dateTime">
          <leaf name="enabled" type="boolean" default="true"></leaf>
          <leaf name="format" type="string" default="%y%m%d.%H%M%S"></leaf>
        </container>
      </container>
    </container>
  </container>
</config>
```

```

        <container name="fcn">
            <leaf name="enabled" type="boolean" default="true"></leaf>
            <leaf name="format" type="string"
default="<CLASS$:<FCNNAME$ ($ARG$)@$LINE$"></leaf>
        </container>
    </container>
    <container name="dest">
        <container name="console">
            <leaf name="enabled" type="boolean" default="false"></leaf>
        </container>
        <container name="file">
            <leaf name="enabled" type="boolean" default="false"></leaf>
            <leaf name="name" type="string" default="mtx-internal.log"></leaf>
            <leaf name="location" type="string" default="./mtxlogs"></leaf>
            <leaf name="mbytes-rollover" type="uint32" default="10"></leaf>
            <leaf name="hours-rollover" type="uint32" default="24"></leaf>
            <leaf name="startup-rollover" type="boolean" default="false"></leaf>
            <leaf name="max-rollover-files" type="uint32" default="10"></leaf>
        </container>
    </container>
    <list name="logitems" key="id">
        <listitem>
            <leaf name="id" type="string"></leaf>
            <leaf name="active" type="boolean" default="true"></leaf>
        </listitem>
    </list>
</container>
</container>
</config>

```

The <list> tag defines the log filters by <componentID>.

The following table describes some of the containers and their leaves.

Table 2: Containers and Leaves

Container	Container Description	Contained Containers	Contained Leaf Description
logging	Contains all logging data types.	<ul style="list-style-type: none"> format dest file <p>Note Also contains list tag logitems</p>	<p>enabled: Boolean that determines whether logging is on or off. Default off.</p> <p>allActive: Boolean that activates all defined logging items for logging. Default off</p>

Container	Container Description	Contained Containers	Contained Leaf Description
format	Contains the log message format information.	<ul style="list-style-type: none"> • componentID • dateTime • type • fcn 	<p>content: String listing data types included in log messages. Includes:</p> <ul style="list-style-type: none"> • \$DATETIMES\$: Include date or time in log message. • \$COMPONENTID\$: Include component name in log message. • \$TYPES\$: Includes message type ("", INFO, WARNING, ERROR) • \$SRCFILES\$: Includes name of source file. • \$SRCLINES\$: Include line number of source file. • \$FCNINFOS\$ Include class::function name from the source file. <p>\$MSG\$: Include actual log message text.</p>
componentID	Name of logged component.	NA	<p>enabled: Boolean that determines if the log message includes the component ID. Default to "true." Value of "false" returns a "" string in log message.</p>
dateTime	Date or time of log message.	NA	<p>enabled: Boolean whether to include date or time information in log message. Default is enabled.</p> <p>format: String of values to include in log message. Format of %y%m%d.%H%M%S.</p>

Container	Container Description	Contained Containers	Contained Leaf Description
dest	Holds destination logger's configuration settings.	console: Destination console. Only one allowed. file: destination file. Multiple allowed.	NA
console	Destination console.	NA	enabled: Boolean that determines whether the console is enabled for logging. Default of "false."

Container	Container Description	Contained Containers	Contained Leaf Description
file	Determines the settings of the destination file.	NA	<p>enabled: Boolean that determines whether the destination is enabled. Default is "false."</p> <p>name: String of the destination log file. Default of "mtx-internal.log"</p> <p>location: String of destination file path. Default at "./mtxlogs."</p> <p>mbytes-rollover: uint32 that determines the length of the log file before the system overwrites the oldest data. Default is 10 Mbytes.</p> <p>hours-rollover: uint32 that determines the length of the log file in terms of hours. Default is 24 hours.</p> <p>startup-rollover: Boolean that determines if the log file is rolled over upon agent start or restart. Default value of "false."</p> <p>max-rollover-files: uint32 that determines the maximum number of rollover files; deletes the oldest file when the max-rollover-files value exceeded. Default value of 10.</p>

Default Config Example

The following is the configuration file with the default installed configuration.

```
<config name="nxos-device-mgmt">
  <container name="mgmtConf">
    <container name="logging">
      <leaf name="enabled" type="boolean" default="false">true</leaf>
      <leaf name="allActive" type="boolean" default="false">>false</leaf>
    <container name="format">
      <leaf name="content" type="string" default="$DATETIME$ $COMPONENTID$ $TYPE$:
```

```

$MSG$">$DATETIME$ $COMPONENTID$ $TYPE$ $SRCFILE$ @ $SRCLINE$ $FCNINFO$: $MSG$</leaf>
  <container name="componentID">
    <leaf name="enabled" type="boolean" default="true"></leaf>
  </container>
  <container name="dateTime">
    <leaf name="enabled" type="boolean" default="true"></leaf>
    <leaf name="format" type="string" default="%y%m%d.%H%M%S"></leaf>
  </container>
  <container name="fcn">
    <leaf name="enabled" type="boolean" default="true"></leaf>
    <leaf name="format" type="string"
default="$CLASS$: $FCNNAME$ ($ARGS$) @$LINE$"></leaf>
  </container>
</container>
<container name="dest">
  <container name="console">
    <leaf name="enabled" type="boolean" default="false">true</leaf>
  </container>
  <container name="file">
    <leaf name="enabled" type="boolean" default="false">true</leaf>
    <leaf name="name" type="string" default="mtx-internal.log"></leaf>
    <leaf name="location" type="string" default="./mtxlogs"/>volatile</leaf>
    <leaf name="mbytes-rollover" type="uint32" default="10">50</leaf>
    <leaf name="hours-rollover" type="uint32" default="24">24</leaf>
    <leaf name="startup-rollover" type="boolean" default="false">true</leaf>
    <leaf name="max-rollover-files" type="uint32" default="10">10</leaf>
  </container>
</container>
<list name="logitems" key="id">
  <listitem>
    <leaf name="id" type="string">*</leaf>
    <leaf name="active" type="boolean" default="false">>false</leaf>
  </listitem>
  <listitem>
    <leaf name="id" type="string">SYSTEM</leaf>
    <leaf name="active" type="boolean" default="true">true</leaf>
  </listitem>
  <listitem>
    <leaf name="id" type="string">LIBUTILS</leaf>
    <leaf name="active" type="boolean" default="true">true</leaf>
  </listitem>
  <listitem>
    <leaf name="id" type="string">MTX-API</leaf>
    <leaf name="active" type="boolean" default="true">true</leaf>
  </listitem>
  <listitem>
    <leaf name="id" type="string">Model-*</leaf>
    <leaf name="active" type="boolean" default="true">true</leaf>
  </listitem>
  <listitem>
    <leaf name="id" type="string">Model-Cisco-NX-OS-device</leaf>
    <leaf name="active" type="boolean" default="true">>false</leaf>
  </listitem>
  <listitem>
    <leaf name="id" type="string">Model-openconfig-bgp</leaf>
    <leaf name="active" type="boolean" default="true">>false</leaf>
  </listitem>
  <listitem>
    <leaf name="id" type="string">INST-MTX-API</leaf>
    <leaf name="active" type="boolean" default="true">>false</leaf>
  </listitem>
  <listitem>
    <leaf name="id" type="string">INST-ADAPTER-NC</leaf>
    <leaf name="active" type="boolean" default="true">>false</leaf>
  </listitem>

```



```

    </listitem>
  <listitem>
    <leaf name="id" type="string">INST-ADAPTER-RC</leaf>
    <leaf name="active" type="boolean" default="true">>false</leaf>
  </listitem>
</list>
</container>
</container>
</config>

```

Change the Log Configuration Using CLI

Since 10.4(2)F, CLIs are available to change the above logging configuration dynamically without restarting the process. These are per agent EXEC. There are not configuration, 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> Example: switch# debug grpc mtx level info	Toggle the logging level: error, warning, info, debug. 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:

```

Log enabled : 1
All active  : 0
Log Level   : Debug

```

```

Log items :
* : 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 : 1
MTX-EvtMgr : 1
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
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 related 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 | errors**

