



## 診断と有用性

この章は次のトピックで構成されています。

- [診断と有用性について \(1 ページ\)](#)
- [コマンドの表示 \(1 ページ\)](#)
- [デバッグログ \(3 ページ\)](#)
- [診断の提案 \(11 ページ\)](#)

## 診断と有用性について

Cisco NX-OSは、Netconf、Restconf、gNMI/gNOI、テレメトリなどの幅広いプロトコルインターフェイスを介して、モデル駆動型プログラマビリティ (MDP) をサポートします。実際、これらのインターフェイスは、共通の基盤となる YANG および DME/ CLI インフラストラクチャを中心に動作します。ユーザーは、一般的に使用される一群のユーティリティにより、動作を診断できます。

## コマンドの表示

このセクションでは、一般的に使用される show コマンドについて説明します。スイッチの実行状態を確認するために使用できます。

表 1: 表示コマンド: 診断と有用性

項目	コマンド	使用法
netconf	<b>show running-config netconf</b>	netconf 構成の表示
	<b>show netconf nxsdk event-history {events   errors}</b>	イベント履歴の表示
	<b>show tech-support netconf</b>	netconf テクニカルサポートの収集
	<b>show netconf internal details</b>	内部状態の確認

項目	コマンド	使用法
	<b>show netconf internal tls service</b>	TLS サーバの状態の確認
	<b>show netconf internal tls session [all] { summary   detail }</b>	現在/過去の TLS セッションのリスト表示
restconf	<b>show running-config   grep restconf</b>	restconf 構成の表示
	<b>show netconf nxsdk event-history {events   errors}</b>	イベント履歴の表示
grpc	<b>show running-config grpc</b>	grpc 構成の表示
	<b>show grpc nxsdk event-history {events   errors}</b>	イベント履歴の表示
	<b>show tech-support grpc</b>	grpc テクニカルサポートの収集
gnmi	<b>show grpc gnmi service statistics</b>	grpc サーバの状態の確認
	<b>show grpc gnmi rpc [all] { summary   detail }</b>	現在/過去の gNMIサブスクリプションのリスト表示
	<b>show grpc gnmi transactions</b>	List gNMI Get/Set
	<b>show grpc internal gnmi subscription...</b>	内部サブスクリプションデータの表示
	<b>show grpc internal gnmi mtx {sessions   statistics subscriptions}</b>	内部インフラログの表示
gnoi	<b>show grpc gnoi service statistics</b>	grpc サーバの状態の確認
	<b>show grpc internal gnoi rpc [all] {summary   detail}</b>	現在/過去の gNOI 接続のリスト表示
openconfig	<b>show running-config openconfig</b>	openconfig 構成の表示
	<b>show openconfig nxsdk event-history {events   errors}</b>	イベント履歴の表示
dme	<b>show system internal dme transaction history</b>	DME トランザクションの確認
	<b>show tech-support dme</b>	DME テクニカルサポートの収集

# デバッグ ログ

このセクションでは、デバッグログを有効にして収集する方法について説明します。

## プログラマビリティ エージェント ログ

Netconf、Restconf、および gRPC エージェントの場合、次の方法でログを収集できます。

- **show コマンド**

これは、エージェント イベントを表示/確認する簡単な方法です。これらのコマンドは、エージェントがクライアント接続とどのように対話するかを確認するのに役立ちます。このログはメモリ内ログであるため、比較的短い履歴しか保持できません。

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

- **ログファイル**

より長い履歴、またはエージェントを無効にした後のログを確認する場合は、**/volatile** ディレクトリに保存されているログファイルを確認します。ユーザーには、スイッチの **bash** シェルにアクセスする権限が必要です。

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

## YANG インフラログ

YANG インフラログは、**/volatile** ディレクトリに保存されます。ユーザーには、スイッチの **bash** シェルにアクセスする権限が必要です。Cisco NX-OS では、Cisco NX-OS dev-ops ロールまたは Cisco NX-OS network-admin ロールに関連付けられたユーザアカウントから **Bash** にアクセスできます。

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

## DME ログ

DME インフラログは、**/nxos/dme\_logs** ディレクトリに保存されます。ユーザーには、スイッチの **bash** シェルにアクセスする権限が必要です。「<https://developer.cisco.com/site/cisco-nexus-nx-api-references/>」を参照してください。

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

## ログ構成の・更

Cisco NX-OSは、パフォーマンスを考慮に入れて、デフォルトでは非常に限定されたログのみを有効にします。

ユーザーは、`/opt/mtx/conf/mtxlogger.cfg`を編集して、詳細さを変更できます。

コンフィギュレーションファイルの構造は次のとおりです。

```
<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="$DATETIME$ $COMPONENTID$ $TYPE$:
$MSG$"></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"></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>
</config>
```

**<list>** タグは、**<componentID>** によってログフィルタを定義します。

次の表では、コンテナとそのリーフの一部について説明します。

表 2: コンテナとリーフ

コンテナ	コンテナの説明	含まれるコンテナ	含まれるリーフと説明
logging	すべてのロギングデータタイプが含まれます。	<ul style="list-style-type: none"><li>• 形式</li><li>• dest</li><li>• ファイル</li></ul> <p>(注) リストタグ <b>logitems</b> も含まれます</p>	<p><b>enabled</b> : ロギングがオンかオフかを決定するブール値。デフォルトはオフです。</p> <p><b>allActive</b> : ロギング用に定義されたすべてのロギング項目をアクティブにするブール値。デフォルトはオフ</p>

コンテナ	コンテナの説明	含まれるコンテナ	含まれるリーフと説明
形式	ログメッセージのフォーマット情報を格納します。	<ul style="list-style-type: none"> <li>• componentID</li> <li>• dateTime</li> <li>• type</li> <li>• fcn</li> </ul>	<p>content : ログメッセージに含まれるデータ型をリストする文字列。 内容 :</p> <ul style="list-style-type: none"> <li>• \$DATETIMES\$ : ログメッセージに日付または時刻を含めます。</li> <li>• \$COMPONENTIDS\$ : ログメッセージにコンポーネント名を含めます。</li> <li>• \$TYPES\$ : メッセージタイプ (""、INFO、WARNING、ERROR) を含めます。</li> <li>• \$SRCFILES\$ : ソースファイルの名前を含めます。</li> <li>• \$SRCLINES\$ : ソースファイルの行番号を含めます。</li> <li>• \$FCNINFO\$ ソースファイルから class::関数名を含めます。</li> </ul> <p>\$MSG\$ : 実際のログメッセージテキストを含めます。</p>

コンテナ	コンテナの説明	含まれるコンテナ	含まれるリーフと説明
componentID	ログが記録されたコンポーネントの名前です。	該当なし	<b>enabled</b> : ログメッセージにコンポーネントIDを含めるかどうかを決定するブール値。デフォルトは「true」です。「false」の値は、ログメッセージに""文字列を返します。
dateTime	ログメッセージの日付または時刻。	該当なし	<b>enabled</b> : ログメッセージに日付または時刻の情報を含めるかどうかのブール値。デフォルトはイネーブルです。 <b>format</b> : ログメッセージに含める値の文字列。%y%am%d%H%M%Sの形式。
dest	宛先ロガーの構成設定を保持します。	<b>console</b> : 宛先コンソール。許可されるのは1つだけです。 <b>file</b> : 接続先ファイルです。複数ファイルが許可されます。	該当なし
コンソール	接続先コンソール。	該当なし	<b>enabled</b> : コンソールのロギングが有効かどうかを決定するブール値。デフォルトは「false」です。

コンテナ	コンテナの説明	含まれるコンテナ	含まれるリーフと説明
ファイル	宛先ファイルの設定を決定します。	該当なし	<p><b>enabled</b> : 宛先が有効かどうかを決定するブール値。デフォルトは「false」です。</p> <p><b>name</b> : 宛先ログファイルの文字列。デフォルトは「mtx-internal.log」</p> <p><b>location</b> : 宛先ファイルパスの文字列。デフォルトは「./mtxlogs」です。</p> <p><b>mbytes-rollover</b> : システムが最も古いデータを上書きする前のログファイルの長さを決定する uint32。デフォルトは 10 M バイトです。</p> <p><b>hours-rollover</b> : 時間単位のログファイルの長さを決定する uint32。デフォルトは 24 時間です。</p> <p><b>startup-rollover</b> : エージェントの起動時または再起動時にログファイルをロールオーバーするかどうかを決定するブール値。デフォルト値は「false」です。</p> <p><b>max-rollover-files</b> : ロールオーバーファイルの最大数を決定する uint32。</p> <p><b>max-rollover-files</b> の値を超えると、最も古いファイルが削除されます。デフォルト値は 10 です。</p>

## デフォルト構成の例

次に、デフォルトでインストールされている設定を含む設定ファイルを示します。

```
<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="`${CLASS}$::${FCNNAME}$ ($ARG$) @$LINE$"></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>
  </list>
</config>
```

```

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

```

## CLI を使用したログ構成の・更

10.4(2)F 以降、CLI を使用して、プロセスを再起動せずに上記のロギング構成を動的に変更することができます。これらはエージェント EXEC ごとに行えます。構成そのものではないため、現在の操作に影響を与えることなく変更できます。

### 手順の概要

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

### 手順の詳細

#### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>[no] debug grpc mtx enable-all</b>	これは、すべてのログを有効にする便利な cli です。
ステップ 2	<b>[no] debug grpc mtx level &lt;level&gt;</b> 例： switch# debug grpc mtx level info	ロギングレベルを切り替えます：エラー (error)、警告 (warning)、情報 (info)、デバッグ (debug) から選択できます。 デフォルトのレベルは <b>info</b> です。

	コマンドまたはアクション	目的
ステップ 3	<b>[no] debug grpc mtx item &lt;item&gt;</b>  例： switch# debug grpc mtx item MTX-EvtMgr	特定の項目のロギングを切り替えます。これは自由形式の文字列で、 <b>show grpc internal mtx debug</b> を使用して使用可能な項目を表示します。

### 例

次の show cli は、現在のロギング構成を表示します。

```
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

```

## 診断の提案

このセクションでは、発生しやすい問題をトリアージするためのいくつかの手順を示します。

## 接続の問題

ユーザーのプログラミングクライアントがスイッチに接続できない場合は、次の点を確認します。

- 実行構成をチェックして、機能が有効になっているかどうかを確認します。
- 個々のエージェントの `show` コマンドをチェックして、サーバが実行されていることを確認します。
- IP/ポートをチェックして、接続がファイアウォールなどによって制限されていないことを確認します。
- クライアントが正しいユーザー名/パスワードを送信することを確認します。
- 証明書ベースの認証を使用する場合は、トラストポイントがスイッチに正しく構成されていること、およびクライアント証明書が一致し、期限切れになっていないことを確認します。

## ネイティブデバイス YANG

読み取り/書き込み操作に関連するネイティブ `openconfig YANG` に問題がある場合は、次の点を確認してください。

- 「書き込み」操作の場合は、DME トランザクションを確認して障害の詳細を確認します。
- 同等の DME REST 要求を送信し、同じ問題があるかどうかを確認します。

## OpenConfig Yang

ネイティブ `openconfig YANG` の読み取り/書き込みに問題がある場合は、次のことを確認します。

- `feature openconfig` が有効になっているかどうかを確認します。
- 公開されている YANG との相違点をチェックして、サポートステータスを確認します。
- `write` 操作の場合は、DME トランザクションを確認して障害の詳細を確認します。

## テレメトリ

テレメトリは、「`feature telemetry`」構成を介して YANG およびその他のデータソースを収集するために使用されます。テレメトリは、「`feature grpc`」を介した `gNMI` 登録にも使用されます。トラブルシューティングの手順は、使用シナリオによって異なります。

## デバッグ ログ

デバッグログは、次の方法で表示できます。

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

**データ/イベント収集の問題 :**

show コマンドで、失敗した、またはスキップされた収集を確認します。

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

**収集時間またはサイズの問題 :**

次の show コマンドを使用して、収集のサイズと時間を確認します。

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

**トランスポートの問題 :**

次の show コマンドを使用して、トランスポートの問題を確認します。トランスポートの問題は、**feature telemetry** シナリオにのみ影響することに注意してください。

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



## 翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。