データ モデルの使用

データ モデルの使用には、次の 3 つのタスクがあります。

データ モデルの取得

データ モデルは、MGBL PIE のソフトウェア パッケージから入手できます。ルータにパッケージをインストールすると、そのパッケージに含まれる特定の機能がインストールされます。Cisco IOS XR ソフトウェアはさまざまなソフトウェア パッケージに分割されているため、ルータで実行する機能を選択することができます。各パッケージには、ルーティングやセキュリティなど、特定のルータ機能のセットを実行するコンポーネントが含まれています。

前提条件：

MGBL PIE ソフトウェア イメージがルータにロードされていることを確認します。

インストール手順については、『*System Setup and Software Installation Guide for Cisco NCS 540 Series Routers*』の「*Perform System Upgrade and Install Feature Packages*」の章を参照してください。

1. データ モデルが `netconf-monitoring` 要求で使用可能であることを確認します。

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
 <get>
   <filter type="subtree">
     <netconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
     <schemas/>
     </netconf-state>
   </filter>
 </get>
</rpc>
```

すべての IOS XR およびシステム管理 YANG モデルが表示されます。

YANG モデルは、**get-schema** コマンドを使用してルータにログインしなくてもルータから取得できます。

スキーマ リストを取得します（データはステップ 2 で使用されます）。

```
<get>
<filter type="subtree">
<netconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
<schemas/>
</netconf-state>
</filter>
</get>
</rpc>
```

ルータ上のすべてのモデルが表示されます。

```
TRACE: 2016/06/13 11:11:42 transport.go:104: Reading from connection
TRACE: 2016/06/13 11:11:42 gnc_main.go:587: Session established (Id: 1009461378)
TRACE: 2016/06/13 11:11:42 session.go:93: Request:
<rpc message-id="16a79f87-1d47-4f7a-a16a-9405e6d865b9"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><get><filter type="subtree"><netconf-state

xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring"><schemas/></netconf-state></filter></get></rpc>
TRACE: 2016/06/13 11:11:42 transport.go:104: Reading from connection
TRACE: 2016/06/13 11:11:42 session.go:117:
Response:
#143589
<rpc-reply message-id="16a79f87-1d47-4f7a-a16a-9405e6d865b9"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<data>
<netconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
<schemas>
<schema>
<identifier>Cisco-IOS-XR-crypto-sam-oper</identifier>
<version>2015-01-07</version>
<format>yang</format>
<namespace>http://cisco.com/ns/yang/Cisco-IOS-XR-crypto-sam-oper</namespace>
<location>NETCONF</location>
</schema>
<schema>
<identifier>Cisco-IOS-XR-crypto-sam-oper-sub1</identifier>
<version>2015-01-07</version>
<format>yang</format>
<namespace>http://cisco.com/ns/yang/Cisco-IOS-XR-crypto-sam-oper</namespace>
<location>NETCONF</location>
</schema>
<schema>
<identifier>Cisco-IOS-XR-snmp-agent-oper</identifier>
<version>2015-10-08</version>
<format>yang</format>
<namespace>http://cisco.com/ns/yang/Cisco-IOS-XR-snmp-agent-oper</namespace>
<location>NETCONF</location>
</schema>
------------<truncated>--------------
```

データ モデルの構造の詳細については、YANG モジュールを参照してください。

次の作業：

ルータとクライアント アプリケーション間の接続を確立するためのプロトコルを有効にします。

プロトコル有効

ルータはプロトコルを使用してクライアントアプリケーションと通信します。ルータおよびクライアント アプリケーションで、要件に基づいて通信プロトコルを有効にします。

- NETCONF
- gRPC

（注） XR で作成された最初の root-lr ユーザのみがシステム管理の最初の root-system ユーザとして同期されますが、連続するユーザは同期されません。XR で作成された連続するユーザはシステム管理には存在しません。そのため、NETCONF または gRPC を介して、連続するユーザによって実行される sysadmin アクセスが必要な操作は失敗します。この制約を克服するには、システム管理で同じ名前のユーザを作成し、適切なグループに割り当てることによって権限を付与します。

プロトコルの詳細については、コミュニケーション プロトコルを参照してください。

SSH プロトコルを介した NETCONF の有効化

NETCONF は、ネットワークを設定するためにセキュア シェル（SSH）転送で使用される XML ベースのプロトコルです。クライアントアプリケーションはこのプロトコルを使用してルータの情報を要求し、ルータの設定を変更します。

NETCONF の詳細については、NETCONF プロトコルを参照してください。

前提条件：

- ソフトウェア パッケージ k9sec package/rpm がルータにインストールされている。
- ソフトウェア パッケージ mgbl package/rpm がルータにインストールされている。
- 暗号キーが生成されている。

NETCONF プロトコルを有効にするには、次の手順を実行します。

1. SSH 接続を介して NETCONF プロトコルを有効にします。

```
ssh server v2
ssh server netconf
netconf agent tty
netconf-yang agent ssh
```

デフォルトのポート番号 830 が使用されています。必要に応じて、1 ～ 65535 内の別のポートを指定できます。

2. セッション パラメータを設定します。

```
router (config)# netconf-yang agent session {  limit value |  absolute-timeout value
 |  idle-timeout value }
```

値は次のとおりです。

- **limit value**：netconf-yang 同時セッションの最大数を設定します。有効な範囲は 1 ～ 1024 です。
- **absolute-timeout value**：絶対セッション ライフタイムを分単位で設定します。指定できる範囲は 1 ～ 1440 です。
- **idle-timeout value**：アイドル セッション ライフタイムを分単位で設定します。指定できる範囲は 1 ～ 1440 です。

3. 統計情報とクライアントの構成設定を確認します。

```
router (config)# do show netconf-yang statistics
router (config)# do show netconf-yang clients
```

例：NETCONF の有効化

```
config
 netconf-yang agent ssh
  ssh server netconf port 830
!
```

例：統計情報を使用した設定の確認

NETCONF 要求の送信後、**do show netconf-yang statistics** コマンドを使用して設定を確認します。

```
show netconf-yang statistics
Summary statistics        requests|             total time|   min time per request|
max time per request|   avg time per request|
other                            0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
    0h  0m  0s   0ms|       0h  0m  0s   0ms|
close-session                    4|       0h  0m  0s   3ms|       0h  0m  0s   0ms|
    0h  0m  0s   1ms|       0h  0m  0s   0ms|
kill-session                     0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
    0h  0m  0s   0ms|       0h  0m  0s   0ms|
get-schema                       0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
    0h  0m  0s   0ms|       0h  0m  0s   0ms|
get                              0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
    0h  0m  0s   0ms|       0h  0m  0s
get-config                       1|       0h  0m  0s   1ms|       0h  0m  0s   1ms|
    0h  0m  0s   1ms|       0h  0m  0s   1ms|
edit-config                      3|       0h  0m  0s   2ms|       0h  0m  0s   0ms|
    0h  0m  0s   1ms|       0h  0m  0s   0ms|
commit                           0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
    0h  0m  0s   0ms|       0h  0m  0s   0ms|
cancel-commit                    0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
    0h  0m  0s   0ms|       0h  0m  0s   0ms|
lock                             0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
    0h  0m  0s   0ms|       0h  0m  0s   0ms|
unlock                           0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
```

```
    0h  0m  0s   0ms|       0h  0m  0s   0ms|
discard-changes                    0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
    0h  0m  0s   0ms|       0h  0m  0s   0ms|
validate                           0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
    0h  0m  0s   0ms|       0h  0m  0s   0ms|
```

例：クライアントを使用した設定の確認

```
show netconf-yang clients
client session ID|   NC version|    client connect time|        last OP time|
last OP type|    <lock>|
22969|                 1.1|         0d  0h  0m  2s|             11:11:24|
close-session|        No|
15389|
```

次の作業：

NETCONF を有効にした後、YANG データ モデルを使用して関連する設定を管理します。

HTTP/2 プロトコルを介した gRPC の有効化

Google 定義されたリモート プロシージャ コール（gRPC）は、オープンソースの RPC フレームワークです。gRPC は IPv4 および v6 アドレス ファミリをサポートしています。

gRPC の詳細については、gRPC プロトコルを参照してください。

前提条件：

- TLS を設定する。

（注） TLS を設定することをお勧めします。gRPC プロトコルを有効にすると、TCP で TLS が有効になっていないデフォルトの HTTP/2 トランスポートが使用されます。gRPC では、すべての gRPC 要求に対して AAA 認証および認可が義務付けられています。TLS が設定されていない場合、認証クレデンシャルはネットワーク上で暗号化されずに転送されます。TLS を有効にすると、クレデンシャルがセキュアで暗号化されることが保証されます。非 TLS モードは、セキュアな内部ネットワークでのみ使用できます。

- ソフトウェア パッケージ mgbl pie がルータにインストールされている。

gRPC プロトコルを有効にするには、次の手順を実行します。

1. HTTP/2 接続で gRPC を有効にします。

```
Router# configure
Router (config)# grpc
```

2. 指定されたポート番号へのアクセスを有効にします。

```
Router (config-grpc)# port <port-number>
```

<port-number> の範囲は 57344 ～ 57999 です。ポート番号が使用できない場合は、エラーが表示されます。

3. コンフィギュレーション モードでセッション パラメータを設定します。

```
Router (config)# grpc{ address-family | dscp | max-request-per-user | max-request-total
 | max-streams | max-streams-per-user | no-tls | service-layer | tls-cipher |
tls-mutual | tls-trustpoint | vrf }
```

値は次のとおりです。

- **address-family**：アドレス ファミリ識別子タイプを設定します
- **dscp**：送信された gRPC での QOS マーキング DSCP を設定します
- **max-request-per-user**：ユーザあたりの同時要求の最大数を設定します
- **max-request-total**：合計同時要求の最大数を設定します
- **max-streams**：同時 gRPC 要求の最大数を設定します。サブスクリプションの上限は 128 要求です。デフォルトは 32 要求です
- **max-streams-per-user**：ユーザあたりの同時 gRPC 要求の最大数を設定します。サブスクリプションの上限は 128 要求です。デフォルトは 32 要求です
- **no-tls**：トランスポート レイヤ セキュリティ（TLS）を無効化します。TLS はデフォルトで有効になっています。
- **service-layer**：gRPC サービス レイヤの設定を有効にします
- **tls-cipher**：gRPC TLS 暗号スイートを有効にします
- **tls-mutual**：相互認証を設定します
- **tls-trustpoint**：トラストポイントを設定します
- **server-vrf**：サーバ VRF を有効にします

次の作業：

gRPC を有効にした後、YANG データ モデルを使用して関連する設定を管理します。

データ モデルを使用した設定の管理

クライアント アプリケーションから、データ モデルを使用してルータの設定を管理します。

前提条件

- ソフトウェア パッケージ k9sec pie および mgbl がルータにインストールされている。
- NETCONF または gRPC プロトコルはクライアントとルータで有効になっている。

データ モデルを使用して設定を管理するには、次の手順を実行します。

1. YANG ツールを使用してクライアント アプリケーションにデータ モデルをインポートします。
2. YANG ツールを使用してデータ モデルの値を変更することによってルータを設定します。

設定可能なデータ モデルの値の詳細については、YANG モデルの構造を参照してください。

例：CDP の設定

この例では、CDP にデータ モデルを使用し、次の表に示す値を使用して CDP を設定しています。

CDP パラメータ	説明	パラメータに必要な値
CDPバージョン	隣接デバイスとの通信に使用するバージョンを指定します	v1
Hold time	受信デバイスが CDP パケットを保持する時間を指定します	200 ms
Timer	ソフトウェアが CDP アップデートを送信する頻度を指定します	80 ms
Log Adjacency Table	隣接関係テーブルに変更を記録します。CDP 隣接関係テーブルのロギングをイネーブルにすると、CDP ネイバーが追加または削除されるたびに syslog が生成されます。	enable

1. ルータから CDP `Cisco-IOS-XR-cdp-cfg.yang` の設定 YANG データ モデルをダウンロードします。データ モデルをダウンロードするには、データ モデルの取得 （1 ページ）を参照してください。
2. 任意の YANG ツールを使用してクライアント アプリケーションにデータ モデルをインポートします。
3. データ モデルのリーフ ノードを変更します。
   - enable（cdp をイネーブルにする）
   - holdtime
   - timer
   - advertise v1 のみ
   - log adjacency

NETCONF を使用した CDP の設定

この例では、CDP のデータ モデルを使用し、NETCONF RPC 要求を使用して CDP を設定します。

```
<edit-config>
  <target>
   <candidate/>
  </target>
  <config xmlns:xc="urn:ietf:params:xml:n:netconf:base:1.0">
   <cdp xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-cdp-cfg">
   <timer>80</timer>
   <enable>true</enable>
   <log-adjacency></log-adjacency>
   <hold-time>200</holdtime>
   <advertise-v1-only></advertise-v1-only>
  </cdp>
 </config>
</edit-config>
```

（注） また、CDPはインターフェイスマネージャを拡張してインターフェイス設定でも設定できます。`Cisco-IOS-XR-ifmgr-cfg` YANG モデルを使用し、インターフェイス設定で CDP を設定します。

gRPC を使用した CDP の設定

この例では、CDP のデータ モデルを使用し、gRPC MergeConfig RPC 要求を使用して CDP を設定します。

```
{
 "Cisco-IOS-XR-cdp-cfg:cdp": {
  "timer": 50,
  "enable": true,
  "log-adjacency": [
   null
  ],
  "hold-time": 180,
  "advertise-v1-only": [
   null
  ]
 }
}
```

（注） また、CDPはインターフェイスマネージャを拡張してインターフェイス設定でも設定できます。`Cisco-IOS-XR-ifmgr-cfg` YANG モデルを使用し、インターフェイス設定で CDP を設定します。

設定のコミット

現在実行中の設定で新しい値を設定するには、設定をコミットします。

また、`confimed-commit` 操作を介して設定をコミットすることもできます。NETCONF および gRPC は confirmed-commit RPC をサポートしています。この RPC では、設定がルータで有効になる前にユーザが明示的に確認する必要があります。この機能は、設定の変更が正しく適用されていることを確認するのに役立ち、管理接続に変更は起こりません。設定変更によって管理接続が失われると、600 秒のデフォルトの `confirm-timeout` 期間後に、設定が以前コミットした設定に自動的にロールバックされます。

設定をコミットするには、</commit> RPC を使用します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <commit/>
</rpc>
```

設定に対して confirm-commit を実行する場合は、次のとおりです。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <commit>
        <confirmed/>
    </commit>
</rpc>
```

confirmed-commit 機能は、<cancel-commit> 操作、<commit> 操作の <confirmed>、<confirm-timeout>、<persist>、<persist-id> パラメータをサポートしています。

```
<rpc message-id="101"
          xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
       <commit>
         <confirmed/>
         <persist>IQ,d4668</persist>
         <confirm-timeout>120</confirm-timeout>
       </commit>
</rpc>
```

confirmed-commit 要求は、次の場合に `Datastore Locked` エラーが発生して失敗します。

- confirmed-commit 操作と confirming-commit 操作の間で別の操作を実行する場合
- 別のセッションでアクティブな confirmed-commit 要求があり、永続 ID が指定されていない場合
- 永続 ID は指定されているが、アクティブな confirmed-commit セッションの永続 ID と一致しない場合

設定をマージする gRPC Confirmed-Commit

gRPC confirmed-commit 要求は、既存の merge-config 操作および cli-config 操作に対して発行できます。この例では、要求は merge-cli 操作に対して作成されています。

```
manageability/ems/client/client -oper merge-config -server_addr="<address>" -json_in_file
 <directory-path>/<file>.json
-confirmed=yes -confirm_timeout=400
enter PID:14917:main.main
emsMergeConfig: Received ReqId 14917, Response '
----------------- gRPC Summary ----------------------
Operation: merge-config
Number of iterations: 1
Total bytes transferred: 126
Number of bytes per second: 374
Round trip throughputs Mbps: 0.002999
Ave elapsed time in seconds: 0.336079
Min elapsed time in seconds: 0.336079
Max elapsed time in seconds: 0.336079
--------------- End gRPC Summary ----------------------
The confirmed commit request should be followed by a confirming commit to make the
configuration permanent:
manageability/ems/client/client -oper commit -server_addr="<address>"
enter PID:14917:main.main
emsCommitConfig: Received ReqId 14917, Response '
----------------- gRPC Summary ----------------------
Operation: commit
Number of iterations: 1
Total bytes transferred: 126
Number of bytes per second: 374
Round trip throughputs Mbps: 0.002999
Ave elapsed time in seconds: 0.336079
Min elapsed time in seconds: 0.336079
Max elapsed time in seconds: 0.336079
--------------- End gRPC Summary ----------------------
```